

UNIVERSITI TEKNOLOGI MARA

**DEVELOPMENT OF TRACK-DRIVEN
AGRICULTURE ROBOT WITH TERRAIN
CLASSIFICATION FUNCTIONALITY**

KHAIRUL AZMI BIN MAHADHIR

These submitted in fulfillment
of the requirements for the degree of
Master of Science

Faculty of Mechanical Engineering

August 2015

AUTHOR'S DECLARATION

I declared that the work in this thesis was carried out in accordance with the regulations of Universiti Teknologi MARA. It is original and is the results of my work unless otherwise indicated or acknowledge as references work. This thesis has not been submitted to any other academic institution or non-academic institution for any other degree or qualification.

I, hereby, acknowledge that I have complied with the Academic Rules and Regulations for Post Graduate, Universiti Teknologi MARA, regulation the conduct of my study and research.

Name of Student : Khairul Azmi Bin Mahadhir

Student I.D. No : 2011266984

Programme : Master of Science (Master in Engineering)

Faculty : Mechanical Engineering

Thesis Title : Development of Track-Driven Agriculture Robot with
Terrain Classification Functionality

Signature of Student : 

Date : August 2015

ABSTRACT

Over the past years, many robots have been devised to facilitate agricultural activities (that are labor-intensive in nature) so that they can carry out tasks such as crop care or selective harvesting with minimum human supervision. It is commonly observed that rapid change in terrain conditions can jeopardize the performance and efficiency of a robot when performing agricultural activity. For instance, a terrain covered with gravel produces high vibration to robot when traversing on the surface. In this work, an agricultural robot is embedded with machine learning algorithm based on Support Vector Machine (SVM). The aim is to evaluate the effectiveness of the Support Vector Machine in recognizing different terrain conditions in an agriculture field. A test bed equipped with a tracked-driven robot and three types of terrain i.e. sand, gravel and vegetation has been developed. A small and low power MEMS accelerometer is integrated into the robot for measuring the vertical acceleration. In this experiment, the vibration signals resulted from the interaction between the robot and the different type of terrain were collected. An extensive experimental study was conducted to evaluate the effectiveness of SVM. The results in terms of accuracy of two machine learning techniques based on terrain classification are analyzed and compared. The results show that the robot that is equipped with an SVM can recognize different terrain conditions effectively. Such capability enables the robot to traverse across changing terrain conditions without being trapped in the field. Hence, this research work contributes to develop a self-adaptive agricultural robot in coping with different terrain conditions with minimum human supervision.

ACKNOWLEDGEMENT

Bissmillahirrahmanirrahim,

In the name of Allah, the Most Benevolent and Most Merciful, praise to Allah S.W.T. Thanks to Allah for giving me His blessing to complete this Master Thesis. I would like to express my deepest gratitude to my supervisor, Dr.-Ing. Low Cheng Yee for an invaluable guidance, consistent advice, sharing his valuable time, encouragement and patience to accept my weakness to complete this project in a long time.

I would also like to acknowledge and express my love and gratitude to my beloved family for their understanding and their endless love through this study the entire period. Thank you to all the endorsement you all have given me all this time. Special thanks to all final year students involved to help me in carrying out experiments and also prepare the final report. Not forgetting to all members of the Laboratory R2 has contributed, directly or indirectly, during the development of this project.

Khairul Azmi

August 2015

TABLE OF CONTENTS

	Page
AUTHOR'S DECLARATION	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	xii

CHAPTER ONE : INTRODUCTION

1.1	Research Background	1
1.2	Problem Statement	2
1.3	Research Objectives	3
1.4	Scope of Research	4
1.5	Research Methodology	6
1.6	Significance of Research	9
1.7	Outline	10

CHAPTER TWO : LITERATURE REVIEW

2.1	Introduction	11
2.2	Agriculture Robotics	11
2.2.1	Wheeled Robot	13
2.2.2	Tracked Robot	15
2.2.3	Legged Robot	16
2.2.6	Reconfigurable Robot	17
2.3	Sensors	18
2.3.1	3-D Imaging	18
2.3.2	Mechanical Sensors	19
2.3.3	Acoustic Sensors	19

2.4	Intelligent Systems	20
2.4.1	Neural Network (NN)	21
2.4.2	Fuzzy Logic	22
2.4.3	Support Vector Machine (SVM)	23
2.4.3.1	Kernel Functions	27
2.4.3.2	SVM for Multiclass Classification Task	28
2.4.3.3	Hierarchical Support Vector Machine	31

CHAPTER THREE : DEVELOPMENT OF TRACK DRIVEN

AGRICULTURE ROBOT

3.1	Introduction	32
3.1.1	Mechanical Design	33
3.1.2	Motor Layout	34
3.1.3	Drive Mechanism	35
3.1.4	Flipper Arm Mechanism	40
3.2	Modelling and Dynamic Simulation of the Mechanical Structure	44
3.2.1	Introduction	44
3.2.2	Design And Analysis	45
3.2.3	Forces	46
3.3	Development of Electronic System	47
3.3.1	Introduction	47
3.3.2	Master Circuit	50
3.3.3	Slave Circuit	51
3.3.4	Accelerometer	53
3.3.5	Encoders	54
3.3.6	Ultrasonic-range finder (SN-LV-EZ1)	56
3.3.7	HMC6352 Compass module	57
3.3.8	Driver	58
3.3.9	Power Distribution	59
3.3.10	Communications	60
3.4	Coordinate System	62
3.5	Experimental Procedure	65
3.5.1	Experiment Process	67

3.6	Simulation And Experimental Results	71
3.6.1	Inner and Outer Velocities of the Specimen via Simulation.	72
3.6.2	Graph Trajectory Motion of the Tracked at Varying Forward Velocities for Simulation and Experimental	74

CHAPTER FOUR : TERRAIN CLASSIFICATION

4.1	Introduction	76
4.2	Principle Solution	77
4.3	Development of the Test Bed	78
4.3.1	Experimental Setup	78
4.3.2	Mobile Track Robot	81
4.3.3	Calibration Process	83

CHAPTER FIVE : RESULTS AND ANALYSIS

5.1	Introduction	85
5.2	Result and Discussion	86

CHAPTER SIX : CONCLUSION AND RECOMMENDATION

6.1	Conclusion	88
6.2	Recommendation	90

REFERENCES	92
-------------------	----

APPENDICES	100
-------------------	-----

LIST OF TABLES

Tables	Title	Page
Table 3.1	List of parameter values of the Track Robot	33
Table 3.2	Comparison between types of Gear	39
Table 3.3	The Parameter Input for Both Side Velocity	68
Table 3.4	The Velocity Required for Each State at Specific Time	69
Table 3.5	List of parameter values of the Track Robot	71
Table 5.1	Accuracy of Classification using Speed Information and Acceleration in Z – Axis	86
Table 5.2	Accuracy of Classification using Speed Information and Acceleration in X, Y and Z – Axis	86

LIST OF FIGURES

Figures	Title	Page
Figure 1.1	Scope of the Research Project	4
Figure 1.2	Flow Activities for the Methodology	6
Figure 2.1	Different type of wheeled robot	13
Figure 2.2	Different type of drive system	14
Figure 2.3	Different type tracked robot	15
Figure 2.4	Different type of legged robot	16
Figure 2.5	Different type of reconfigurable robot	17
Figure 2.6	Comparison between Standard SVM Binary Classifications and Multiclass SVM Classification	24
Figure 2.7	Separating Hyperplane in the SVM between Two Data Sets	29
Figure 2.8	The effect of soft margin constant C	30
Figure 2.9	Hierarchical Support Vector Machine	31
Figure 3.1	Overall Mechanical Design of the Track driven robot	33
Figure 3.2	Position Drive Motor and Flipper Arm Motor	34
Figure 3.3	Design Architecture for the Drive Mechanism and Gear Ratios	35
Figure 3.4	Diagram of Track-Driven Robot	36
Figure 3.5	Illustration of the worst case scenario	37
Figure 3.6	Agriculture Track Robot climbing obstacle	40
Figure 3.7	Automotive Dc Motor for the Flipper Arm	41
Figure 3.8	Complete flipper arm assembly with Helical Gearbox	43
Figure 3.9	Property of Aluminium Alloy	45
Figure 3.10	Directional Deformation X, Y, Z Axis based on The Applied Force	46
Figure 3.11	The Brain of the Track Robot System	47
Figure 3.12	Electrical Design Architecture	48
Figure 3.13	Inter-integrated Circuit (I ² C) for the Communication	49
Figure 3.14	Master circuit on the Agriculture Robot	50
Figure 3.15	Slave Circuit using Fritzing Software	51

Figure 3.16	Three Accelerometer from SparkFun Electronics	53
Figure 3.17	Encoder used during the experiment	54
Figure 3.18	Quadrature phase of Encoder	54
Figure 3.19	Ultrasonic-range finder (SN-LV-EZ1)	56
Figure 3.20	Magnetic compass (HMC6352)	57
Figure 3.21	Duty cycle of Pulse Width Modulation (PWM)	58
Figure 3.22	Power Management Design	59
Figure 3.23	PlayStation Controller	60
Figure 3.24	Transmitter and Receiver Module For 2.4Ghz Wireless System	60
Figure 3.25	ZigBee Module: Transmitter And Receiver	61
Figure 3.26	Coordinate system for tracked vehicle analysis.	63
Figure 3.27	The position of ultrasonic-range finder during the experiment.	65
Figure 3.28	Position of both sensors during the experiment.	66
Figure 3.29	Spacious space for experiment	67
Figure 3.30	Wooden wall sets perpendicularly to the Track Robot	67
Figure 3.31	Robot settings before the experiment starts.	68
Figure 3.32	The switch button is pressed to start the motion	69
Figure 3.33	Prototype of Small Scale Track Robot	71
Figure 3.34	Inner and outer velocities of the specimen via simulation.	72
Figure 3.35	Inner and outer velocities of the specimen via experiment	72
Figure 3.36	Trajectory of the tracked robot at, (a) 0.534 m/s via simulation, (b) 0.356 m/s via simulation and (c) 0.178 m/s via simulation	74
Figure 3.37	Trajectories of the tracked vehicle at varying initial forward velocities via experiment.	75
Figure 4.1	Principle Solution of a Track Agriculture Robot with Terrain Classification Functionality	77
Figure 4.2	Test Bed Setup and Mobile Robot	78
Figure 4.3	Different type of Terrains	79
Figure 4.4	Example of data taken using IMU on a different type of terrain	80

Figure 4.5	Testing and Data Acquisition	81
Figure 4.6	IMU position during the calibration process	83
Figure 6.1	Future Improvement for Terrain Classification	91

LIST OF ABBREVIATIONS

Abbreviations

IMU	Inertial Measurement Unit
CPU	Central Processing Unit
RPM	Revolution Per Minute
τ	Torque
F	Force
d	Distance
DC	Direct Current
C_G	Centre of Gravity
m	Mass
g	Gravity
μ	Coefficient of Friction
I/O	Input and Output
PWM	Pulse Width Modulation
θ	Yaw Angle
$\dot{\theta}$	Yaw Rate
V	Linear Velocity of the origin of moving axes
β	Side Slip Angle
$\dot{\beta}$	Side Slip Ratio
φ	Directional Angle ($\theta - \beta$)
V_x	Forward Velocity
V_y	Lateral Velocity
a_x	Forward Acceleration
a_y	Lateral Acceleration
R_C	Radius of Curvature
C	Soft Margin
D_T	Training Data
x	Direction of the heading robot in X axis
y	Direction of the heading robot in Y axis

CHAPTER ONE

INTRODUCTION

1.1 RESEARCH BACKGROUND

In the last decades, an increase number of robotic systems has been developed to assist human workers in agricultural activities, for instance, robot-assisted methods for fertilization, spraying, fruit harvesting and transferring process [1][2][3]. Recent advances in software have allowed the robots to possess the ability to adapt to their environment [4] by learning from the data about the surrounding. One of the approaches is the deployment of machine-learning techniques [5]. In an agricultural field, the terrain condition has an affects to the performance of the robot in carrying out a task. Gravel, for instance, produces high vibration to robots traversing on such surface. In this work, two machine-learning techniques based on support vector machine (SVM) are proposed as a learning algorithm to distinguish different terrain conditions in an agricultural field. To evaluate the effectiveness of the algorithm, a track-driven mobile robot is embedded with a MEMS accelerometer used to measure vibration data which is then analyzed and classified using SVM. Having knowledge about the terrain condition, the control of the motor drive can be adapted to produce the thrust required for the mobility of the robot when traversing on changing terrain conditions in the field.

1.2 PROBLEM STATEMENT

The rapid growth of the world population poses a threat to the sustainability of food supply. The traditional agriculture industry is labor intensive. Manual process, such as weeding and harvesting process limit the production of quantity crops. Agriculture robotics plays an important role to optimize the production of crops and ensure the sustainability of food supply in the future. Various types of robot are being developed to accomplish labor intense task such as planting, spraying and harvesting. The basis for the feasibility of such agriculture robots is the ability to traverse across various terrain conditions. This is due to the fact that agriculture terrains can be quite challenging even for human to navigate off road vehicles. With such knowledge on the terrain, the robots can improve its performance stability and path planning for autonomous operation. The first step in enabling autonomous operation is implementing proprioceptive sensors such as accelerometer to collect data for the purpose of terrain classification because the current technology using image processing for terrain classification suffer few drawbacks such as it must have enough light for the system to work properly. In this work, vibration-based terrain classification is proposed. With the help of machine learning techniques, an autonomous robot is able to know the terrain condition and adapt its behavior accordingly for a safe operation over an unknown terrain.

The first phase in building an autonomous agriculture robot is to solve the main objectives of the research which to simulate the behavior of the track-driven robot and to understand the interrelation between the track – terrain when it traversing. Then the robot is implemented with a learning machine to enable the robot distinguish between terrains.

1.3 RESEARCH OBJECTIVES

A number of objectives in this work allow the understanding the relationship track robot and the classification process. These are as follows:

1. Kinematic behavior of a track-driven agriculture robot

In order to understand the behavior of a track-driven robot, a simulation of the kinematic behavior of a track-driven agriculture robot is needed. The motion of the robot is simulated and compared to the lab control environment.

2. Experiment Study on the Track-Terrain Interaction.

In a real agriculture field, a robot is exposed to vibration of the terrains. An experiment is needed to be conduct on the track-terrain interaction when an agriculture robot is traversing on the sand, soil and gravel. Then the data in term of vibration of three type of terrain will be recorded.

3. Vibration-based terrain classification using SVM and HSVM.

The classification algorithm will be considered using a Support Vector Machine and Hierarchical Support Vector Machine. Both algorithms will be subjected to training and test pattern of vibration of three types of terrain. Then both algorithms will be compared in term of accuracy

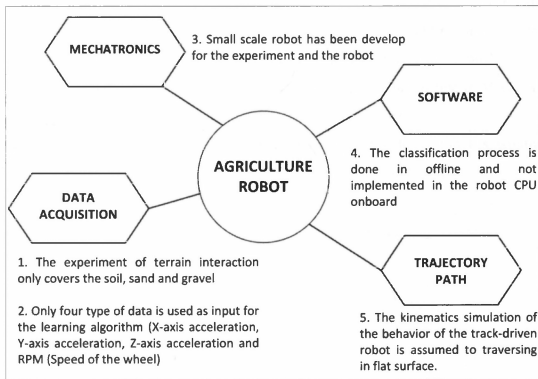
4. Development of a Track-Driven Agriculture Robot

Before the experiment is conducted, a robot is needed to aid the experiment process. Both mechanical and electrical is developed which include the sensor and the track system.

1.4 SCOPE OF RESEARCH

In the background of research provide the necessary information to set up the scope of research. A thorough review of the literature gave insight on what is the challenges and limitation of the agriculture robot behave in the field .The agriculture robot research is in the initial phase, the scope covers four interrelated area shown in Figure 1.1

FIGURE 1.1
Scope of the Research Project



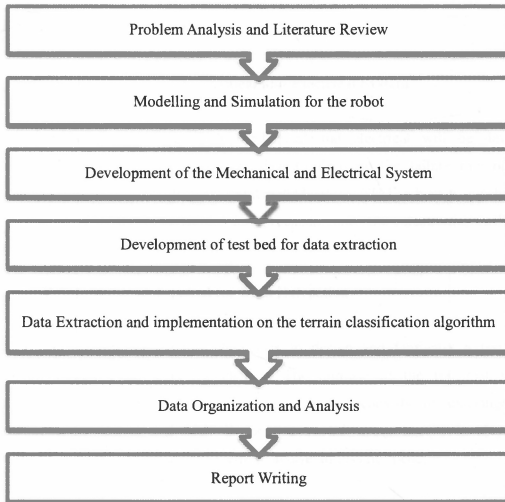
Due to wide scope in the autonomous agriculture robotics, the scopes are limited and listed as follows:

- During the data acquisition process only four types of data is used as input for the learning algorithm (X-axis acceleration, Y-axis acceleration, Z-axis acceleration, and RPM (Speed of the wheel). In the experiment of the track – terrain interrelation covers only the soil, sand and gravel. Only three types of terrain is covered in the experiment which to create a control environment in the laboratory.
- In the mechatronics section, there are two robots are developed in term of mechanical and electronics for both real scale and small scale robot. Only the small scale robot is used during the experiment and the robot with the flipper arm is the concept for future autonomous agriculture robot.
- In term of software, the classification process is done offline in MATLAB and not implemented in the robot main CPU onboard.
- The simulation of kinematic behavior of the track-driven robot is assumed to traversing in flat surface for the trajectory path section.

1.5 RESEARCH METHODOLOGY

The proposed research begins with discussing the problems and the literature review on the system and technology available. Then modeling and simulation is done on the mobile robot to study the output kinematics using MATLAB and ANSYS software to understand the behavior of the robot. After the modeling and simulation, the fabrication process for both mechanical and electrical is proceed for the robot. The test bed then is fabricated which consist of different type of terrain for data extraction. The extracted data is implemented on the classification algorithms for terrain classification process which shown in Figure 1.2. Next step leads to data organization and analysis and end up with report writing.

FIGURE 1.2
Flow Activities for the Methodology



i) **Problem Analysis and Literature Review**

Review of current research will be in agriculture robotics as well as computational intelligence of technical systems and the limitations of the computational intelligence will be taken into consideration.

ii) **Modeling and Simulation for the Track Robot**

The mathematical modeling for the track agriculture robot is done based on the previous researcher. The model simulated on the kinematic simulation of the track driven robot and is compared to a small scale robot from actual system. The kinematics for the robot is done using MATLAB Simulink based on few assumptions.

In this flow includes the design of the mobile robot using CATIA and simulation using ANSYS software to understand the expected design output in term of Forces on three axes (X, Y and Z) on the robot structure.

iii) **Development of the Mechanical and Electrical System**

During this process, both mechanical and electrical will be develop based on the modeling and simulation data to decrease the failure rate of the robot. The mechanical structure is fabricated at UiTM advance machining lab using 3 axis CNC machine for precision and the electronics will be develop in the Robotic Research Lab.

iv) **Development of Test bed for Data Extraction**

The motivation for this chapter is to design and fabricates a test bed for data extraction for next the chapter. The purpose of the test bed is to create a control environment for the robot which consists of exchangeable plates of sand, gravel and soil. Data is extracted using accelerometer from SparkFun Electronics and encoder in term of accelerations and speed.

v) **Data extraction and implementation of Terrain Classification Algorithm**

The aim in this phase is to evaluate the data from the test bed and Computational Intelligence which includes Support Vector Machine and Hierarchical Support Vector Machine is attempted. The objective function for the algorithm is to construct the optimal separating hyper plane to distinguish between the data sets. In higher dimensional feature space, kernel function is used to construct the mapping for the Support Vector Classification.

vi) **Data Organization and Analysis**

Both results for the simulation and small scale robot are compared. Such comparison provides useful information for the development of the real size track robot. For the classification result from SVM and HSVM is compared and used for future integrated in the robot.

vii) **Report Writing**

In the last phase, all the data from the problem statement until the analysis will be compile and concluded accordingly to the format required by the Universiti Teknologi MARA standard.

1.6 SIGNIFICANCE OF RESEARCH

An autonomous robot is an exciting and challenging research for reasons. The first reason is the provide a computer to be able to sense real world properties such a terrains into an learning machine or intelligent machine to detect pattern, identity features and navigates thru terrains. Thus Support Vector Machine is chosen and compared with Hierarchical Support Vector Machine in term of classification accuracy.

To build a track-driven agriculture robot, a particulars understanding of the interaction between track and terrain is needed in term of kinematics. The knowledge is needed to understand the behavior of the tracks during motion and how the tracks behave during turning.

In the development of autonomous robot, a terrain classification is a compulsory to enable the robot to learn about the surrounding terrain. This knowledge then is used by the robot to navigate thru the terrain and avoid being trapped in the terrain. This kind of situation is unwanted when the robot is picking ripe fruits thus affecting the quantity of the crops can be collected.

1.7 OUTLINE

This thesis is divided into six chapters as summarized below. Chapter 2 is the literature review which describes the past research that has been done in this field and begins with findings on the artificial intelligence and current technology used by others. This chapter provides the explanation on artificial intelligence. Chapter 3 aims to present the development of the track-driven agriculture robot during the prototype development. This also includes the mechanical, electrical development and also simulation of the track-driven robot which done using MATLAB. In this chapter the result is of the simulation compared to small scale robot from actual system.

Chapter 4 contains the learning algorithm of the robot which elaborate for both Support Vector Machine and Hierarchical Support Vector Machine and which involve the fundamental theory of the SVM and the learning model for the classification. The algorithm uses four types of kernels which include the Linear Function, Quadratic Function, Polynomial Function and Radial Basis Function. The data collection is done in a controlled environment which imitator the real agriculture terrain. The collected data then will be used for terrain classification. This chapter will briefly discuss on the test bed design and procedure during the data extraction. Chapter 5 discussed on the result and analysis of the classification obtained from the Support Vector Machine and Hierarchical Support Vector Machine. Chapter 6 summaries all the work in the thesis is summarized. The current work is discussed with summarized results and recommendation on future works are proposed in order to overcome the problem and limitation in support vector machine implementation.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter discusses the findings on a different type of mobile robot available and the intelligence system which leads to a terrain classification. This chapter provides the explanation on artificial intelligence used in the research which is Support Vector Machine and Neural Network. All the reviews based on journals, books, and online articles related to the project.

2.2 AGRICULTURE ROBOTICS

Autonomous navigation technologies for off road terrain are rapidly researched and developed [6]. It is one of the crucial elements needed in agriculture robotics development. This technology not only being employed in military but also for normal civilian purposes for wide-area environment monitoring [7] and new terrain explorations. The challenges for such system are to develop the ability to sense and know the environment and manipulate the information for feedback control. In the field of research, there are many type of robot developed and available for commercial used. There are different a categories of mobile robot such as wheel robot, tracked robot, legged robot, aerial robot, underwater robot and reconfigurable robot

Many robotic systems have been developed to ease the work of human in agriculture which is labor intensive in nature. The aim of developing autonomous robots for agriculture automation is to minimize human supervision during tasks execution such as harvesting or crop care.

In agricultural automation, robots is equipped with a computer vision system to perform visual navigation [8]. For example, a low-cost robot is equipped with a vision control system to provide a visual navigation for fertilization and spraying artificial pollination [9] in a greenhouse environment. Computer vision systems are also installed in a robot to guide it

to travel between the crop rows [10] and to perform automatic recognition on the fruit conditions before harvest [11] or for fruit grading [12]. On the other hand, there is also research on fusing the agricultural robots with machine-learning techniques [5][13]. For example, a harvesting robot [14] is installed with a statistical machine-learning method to recognize the maturity of apples. A computer vision system is integrated with artificial neural networks to perform leave image classification for sunflower crops [15]. In a different approach, a normal CCD camera [16] is used for the harvesting process. The robot is equipped with cutting tools and a camera with the capability to differentiate between ripe and unripe crops. Some researcher uses more than one sensor [17] to increase the classification rate during the harvesting process. The system uses both binocular-vision and sonar to classify using hue and saturation of color histogram during the harvesting operation. An interesting idea came from a researcher [18] which uses a robot for weeding process. The robot is designed to be able to adapt the speed based on the size of the paddy field and the soil condition.

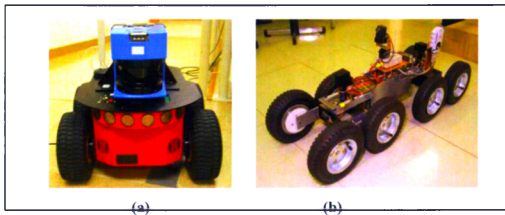
Terrain classification using a computer vision based system [19] is popular. A system developed by [20] is able to classify terrain using images provided by a single camera and it consumes less power compared to the laser range finder. In [21], a monocular camera is used to provide knowledge about the terrain.

Iagnemma and Dubowsky [22] measured the vibration profile of a low speed rover running over different terrains. The vibration is measured by an accelerometer in three axes (X, Y, and Z). Each terrain produces a characteristic profile which can be used for classification. Compared to the vision based system, this method consumes less energy and computation time. Further, the vibration based approach does not depend on a good lighting condition which is necessary for a vision based system [23]. A similar research on terrain classification has been done using a crawling robot [24] equipped with an Inertial Measurement Unit (IMU).

2.2.1 Wheeled Robot

The wheeled robot usually used in the research field [25] due to the simpler design than the legged or tracked robot. The robot usually design and develop for flat movement and not for rough terrain (low friction are or rocky surface). There is no limit numbers for the wheel robot in the development which based on the application example from Henan University using two wheeled robot [26] with natural instability body mimicking the inverted pendulum, other researcher uses three wheeled [27] with the Modular Universal Unit (MUU) perform as pitch, yawing and roll. This is achievable with passive rollers at center of the cylindrical shell and forming the motion for the robot. The four wheeled robot [28] is the most preferred by researcher due to ease on control and low cost to develop. Most of the cases use the skid steer [29] which has higher steering capabilities. In rare cases, researcher uses eight wheeled robot shown in Figure 2.1(b) for climbing stairs and uneven terrain. Higher level of controller is needed to control the wheels for optimizing the moving efficiency and speed.

FIGURE 2.1
Different type of wheeled robot (a) P2 –AT robot [29], (b) Octal Wheel [30]

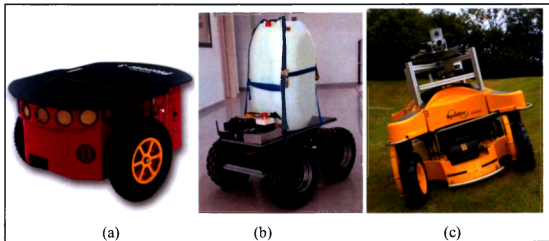


There are many type of steering system in the development of wheel robot for example skid steer drive, differential drive, and synchronous drive. The common drive type used in the research field is the skid steer drive system [31] which uses a separate motion of the wheel as the steering system and popular due to mechanical simplicity and low cost for development. The skid steer drive system usually used in

the tank and works when the right and left wheel is driven independently with different speed and resulting in the robot to skid on the surface. This system capable of archiving higher turning radius compared to other type of robots which make it highly maneuverable depending on the terrain. The differential drive system [32] mechanism works when two set of motor is control independently and easy to be used by beginner as shown in Figure 2.2. Such drive system with a different friction and motor profile resulting in difficult for a straight line movement. For the synchronous drive system [33], the motion and direction is made possible with the sets of motor system mechanically coupled which move in the same speed and direction.

FIGURE 2.2

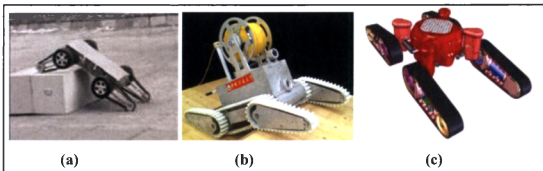
Different type of drive system (a) Pioneer 3-DX [34] with differential drive system, (b) Quadriga robot [35] with skid steer drive system, (c) Spider robot with synchronous drive system [33]



2.2.2 Tracked Robot

In the real world application the usage of the mobile robot is limited due to capability of the robot to traverse in the urban environment or agriculture field. Over the years, rapid development of track robot has been made to overcome such challenge. The platform uses track compared to wheel for motion and navigate across obstacle. To overcome the locomotion problem, a platform called AZIMUT [28] has been develop by University of Sherbrooke shown in Figure 2.3. The platform uses four track with independent articulation with a three degree of freedom (DOF) on the joint. The freedom allows higher flexibility and adaptability in the movement. A different approach is presented by Robotic Department of Ritsumeikan University using a hybrid [36] track mechanism during the operation. The hybrid track mechanism use a fixed track mechanism and transformable track mechanism which more adaptive to uneven or bumpy terrain. The basic idea of a track drive system is to use sensors for closed loop feedback when traversing across terrain and more complex algorithm is needed for learning about the environment. However with self-adapting mechanism the mobile track robot able to efficiently adapt over a terrain with different configuration without sensor feedback thus reducing the time lag. Similar research has done in agriculture field which is used in the paddy field. The researcher uses Laser range finder and Inertial Measurement Unit [37] as path finder and to stabilize the robot during traversing on the irregular paddy surface.

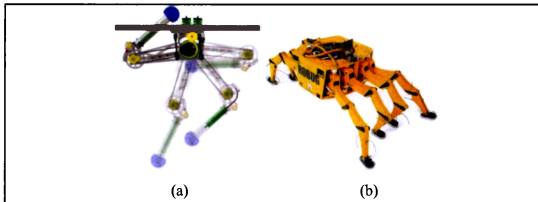
FIGURE 2.3
Different type Tracked Robot (a) MOBIT [38] from the Beijing Institute of Technology (b)
CUMT- III robot [39], (c) AZIMUT track robot from Ritsumeikan University [40]



2.2.3 Legged Robot

Legged locomotion is common in the nature and known to have better adaptability when walking in rough nature terrain compared to wheel or track robot. This has motivated researcher to develop a legged robot which able mimic animals or humans. A researcher in Waseda University has developed a legged rat robot [41] that possesses body and leg comparable to real rats. The robot has 3 degree of freedom (DOF) with two active and one passive on each legs. The robot uses four legs mimicking the real rats performing task in a most natural way like pushing levers. Other non-conventional motion is the hopping motion which required a complex motion control [42] and to avoid reaction force from damaging the robot actuator. The challenge of creating legged robot to move efficiently in unstructured terrain has inspire Portsmouth University to develop an eight legged [43] robot mimic the motion of terrestrial crab in Figure 2.4.

FIGURE 2.4
Different type of Legged Robot (a) ScarlETH [44] with two legs (b) Eight Legged robot [43].

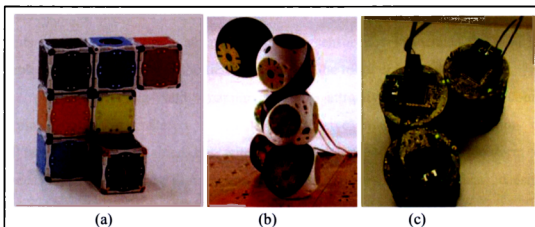


The actuator uses a pneumatic drive system to power up the joint with high power to weight ratio on each limb for crossing surfaces and crawl using insect gait. The insect gait or locomotion has higher stability in motion compared to humans or mammals using dynamic stability [45] like an inverted pendulum motion. The low center of gravity of an insect uses a static stability [46] with at least three legs contact with the ground to maintain balance.

2.2.4 Reconfigurable Robot

The configurable robot [47] is more flexible compared to other mobile robot. The design allows the mechanism of the robot to traverse on the unstructured environment and external condition and optimizing the task given. The M-Block robot from Computer Science and Artificial Intelligence Lab MIT is design with nobility for self-assembly and self – reconfigurable [48] which uses magnetic bond and angular momentum actuator as shown in Figure 2.5. The actuator is coupled with flywheel employing high torque motion breaking the bond between the modules generating motion for the robot configuration. Other researcher uses dynamic connection [49] for motion and self-configuration. The dynamic connector between the robots module enable of changing in structure based on the specified task or in this case intelligent furniture.

FIGURE 2.5
Different Type of Reconfigurable Robot. (a) M-Block from MIT Labs,
(b) Roombots [50] , (c) Planar Catoms [51].



2.3 SENSORS

For agriculture robot, the capability to interact with changing environment is the important key aspect in construction of an autonomous robot. In order for successfully constructing such behavior based architecture [51] is to have capacity to react to sensory inputs for an instance using IR distance sensor in obstacle avoidance.

2.3.1 3-D Imaging

Researcher from Institute of Technology Pasadena [52] uses stereo vision in the rover to detect potential terrain vulnerabilities before traversing across it. The rover navigates using stereo vision by plotting a local map of surrounding area and analyzed for most effective travelling path.

In contrast to stereo camera, LIDAR scanner is an optical remote sensing technology which measure the distance by revealing the object with laser pulse thus creating an image of the object. The sensor is used by the University of Applied Sciences Osnabrück [53] for detecting obstacle or plants in the fields. Such advantages of the sensor not influenced by sunlight enable 24 hours operation in the field maximizing the robot efficiency.

Visual sensing and mapping offers attractive benefits for mobile terrain robot. Using Stereo vision [54] extensive amount of information on the terrain can be obtained but with few drawbacks of complexity in term of processing power and illumination of the terrain object surface causing repeated patterns.

2.3.2 Mechanical Sensors

In a different perspective, University of Technology Thonburi Thailand [55] uses Inertial Measurement Unit or IMU for acquiring data of the terrain which has been traverse by the robot. The concept of the IMU in the system works completely differently than a LIDAR or optical sensor which detects and classified terrains before the robot traversing on it. The IMU is a mechanical sensor which measured the acceleration when traversing on the terrain and then classified it afterwards.

Other method of terrain classification uses system can be called “sensing by feeling” since it uses an internal sensors to determine the terrain surface. The concept is the same as human navigating a car on the rough off road terrain using “feeling of touch” to feel the road condition rather using navigation data to adjust the steering or speed. Such simplicity comes with a few drawbacks such as noise either from the sensor or the mobile robot that can reduce the efficiency of the terrain classification.

2.3.3 Acoustic Sensors

On the other hand, from Ocean System Engineering Research Department Korea [56] uses acoustic sensor or sonar in their underwater multi legged robot for plotting an image of sea floor with low visibilities or in strong murky water.

2.4 INTELLIGENT SYSTEMS

Humans have always been fascinated with their own capabilities to think and to learn. They tend to imitate these capabilities by using different approaches. Such methods are implemented using computer programs to imitate these intelligent based on special algorithms. Some would ask what are intelligent robots or what it will be able to do or imitate. According to Robotic Industrial Association (RIA) [51] a robot is a re-programmable, multifunction and designed to perform a variety of tasks.

An intelligent machine [57] with the ability of executing assignment by themselves without human intervention is called autonomous robots. A set of sensors with processing capabilities are needed to enable a robot to manipulate its actuators for autonomous activities. Autonomy is a system capable to run in the real-world environment without external control within a period of time.

Recent challenges of implementing this intelligent machine are introduced in DARPA Grand Challenge [58] on March 2004. It is a challenge proved to be a difficult task for 19 unmanned vehicles through a harsh route 142 miles across the Mojave Desert.

2.4.1 Neural Network (NN)

One of the famous artificial intelligence used by many researchers is the Neural Network. It is a learning machine used to predict the group belonging for a data. The learning machine is a simplified model of a biological neuron system [59] which contained high interconnected neural of computing elements with the ability to learn and gaining knowledge for use. The Neural Network is a supervised learning which need an input from the user to “label” the data for the classifications logic to work compared to the unsupervised learning has the capabilities to categories the data based only from the raw data inputs. According to Iran University of Science & Technology (IUST) in their research, Neural Network has the capabilities to classify cardiac arrhythmias [60] with the 100% accuracy using Multilayer perceptron (MPL). The input from the HRV signals obtained from the databases and then is classified in to four type of life threatening cardiac arrhythmias.

In the off road terrain, a critical algorithm is needed to guide an autonomous robot to safety. Based on the research by Agency for Defense Development Korea which uses Neural Network Classification with Speed Up Robust Features (SURF) [6] for off road terrain classification. The method uses supervised learning which extracts features distinguish the ground truth image and producing higher classification rate compared to the wavelet classifications.

A comparative study of classifiers for Thalassemia screening attested SVM a better performance than K-Nearest Neighbor [61]. SVM and K-Nearest Neighbor has been used [62] for large scale hierarchical text classification and conclude that k-NN performs better than SVM. The classification of sonar signals [63] is done using Neural Networks and Decision Trees and the results shows, that the Neural Network clearly outperforms various Decision Tree classifiers.

In recent years the use of different classifiers in robotic applications has been studied. In a robotic soccer formation [64], comparison between SVM, k-Nearest Neighbor, Naïve Bayes and Neural Networks is done and conclude that SVM performs best when the test set is independent. Few [65] manage to achieve 100% accuracy using a Neural Network to recognize scenarios based on information provided by ultrasonic and light sensors.

2.4.2 Fuzzy Logic

The fuzzy traversability index [66] has been used by Howard [67] as the rule base for quantifying the travel of a terrain by a mobile robot which acquired from image data to measure the terrain classifications. Based on the algorithm [68] the terrain can be classified into four types which is terrain roughness, hardness, slope and discontinuity. Roughness can be divided in to two; indicating surface irregularity and coarseness which also can be asserted as rough or smooth of the surface. Hardness is to measure the hardness of the surface that can influence traction of a mobile robot. Slope is measure based on the incline or decline of the mobile robot to the ground that can be classified as steep, flat or sloped. The discontinuity is to representing terrain such as cliffs or ravines.

2.4.3 Support Vector Machine (SVM)

Few algorithms can be used for the terrain classification such as Artificial Neural Network (ANN), State Vector Machines (SVM) or Fuzzy Logic. Many researchers use Support Vector Machines (SVM) as their classification algorithms [23] because of high generalization capability compare to other method. Massachusetts Institute of Technology (MIT) [69] developed an algorithm to be implemented on planetary rovers to reduce human supervision using two parameters which is determine soil shear strength based on the internal friction angle and cohesion of the soil [70].

Some of the researcher uses probabilistic modeling technique [71] for high-speed rough terrain mobile robot. They results shown in well-known terrain the mobile robot can accurately predict the performance, however in unknown terrain the accuracy declining cause by combination of terrain complexity and imprecise terrain knowledge.

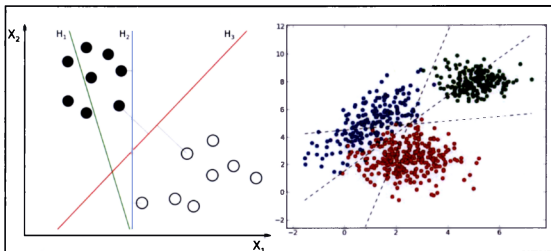
Classification algorithms have been applied to a large amount of real-world problems and much research was done to compare the performance of different classifiers. The classifiers such as Naïve Bayes [72], Decision trees and SVM is compared on 13 binary datasets from the UCI repository and their results show that there is no statistical difference between them. The evaluated classification error of SVM [73] compared to 16 other methods (e.g. Decision Trees, Nearest Neighbor, Neural Network). Their results show a good performance of SVM in most cases, but an overall superiority cannot be confirmed. During the experiment [74][75] both used SVM and Naïve Bayes for their work related to a determined real-world problem. For emotion Classification SVM and Naïve Bayes yielded nearly equal accuracies, while SVM outperformed Naïve Bayes in predicting the Arboviral Disease-Dengue.

Research has been done in the field of terrain classification performed by different types of robots. In the application of 1-legged robot [76] both SVM and Neural Network were applied to classify terrains. Considering only ground reaction force data accuracies up to 78% were obtained for both classifiers. In different research the classification of terrain are based on torque and power consumption by using a modular snake-like robot [77]. A closely related research to our work was done by [78], who have compared the performance of different classifiers for

vibration-based terrain classification. Their results showed that SVM outperformed other classifiers for this application, namely Probabilistic Neural Network, Brook's Method, k-Nearest Neighbor, Decision Trees and Naïve Bayes. They also investigated on the classification results for different robot speeds, concluding that data collected at lower speeds were more difficult to classify and mixed datasets had a negative impact on the classification performance when compared to the results on individual velocities.

The Support Vector Machines is a supervised learning model consists of training data input and output which is used to for the either for classification or regression analysis. Standard Support Vector Machine (SVM) is developed to solve binary classification or Dichotomic classification (two classes only). A problem occurs when more than two classes need to be classified and is resolved by break it down to a several binary problems as the standard Support Vector Machine shown in Figure 2.6.

FIGURE 2.6
Comparison between Standard SVM Binary Classifications and Multiclass SVM Classification



The motivation in using Support Vector Classification is to find the optimal separating hyperplane that is believed to be optimal separated if the space between vector to hyperplane is optimal and without separation errors. Classification or which is also referred to as supervised learning in the literature, is the task to categorize a given instance into one of several previous known classes. Algorithms that fulfill this task are called classifiers. A classification algorithm involves two phases: training and application phase. During the training phase the algorithms tends to learn a model based on a given training set consisting of labeled instances

In the application phase the classifier assigns the most likely class to a new observation based on the learned information. A lot of different classifiers have been developed in the past decades: i.e. Naïve Bayes, K-Nearest Neighbor, Decision tree, Neural network and SVM.

(a) Naïve Bayes

This classification algorithm considers the probability for each class c_i given the observed attributes A of the requested instance and assigns the class with the highest probability to it. According to Bayes Theorem these probabilities can be calculated as

$$P(c_i|A) = \frac{P(A|c_i) \cdot P(c_i)}{P(A)}$$

Naïve Bayes classifier assumes all attributes to be statistically independent, which simplifies the calculation of $P(A|c_i)$. As a consequence only the probabilities $P(c_i)$ and $P(a_i|c_i)$ for all classes c_i and attribute realization a_i is needed. The training phase is used to estimate them.

(b) K-Nearest Neighbor

To determine the class of an unknown instance the k-Nearest Neighbor classifier considers similar instances from the training set. As instances with n attributes can be interpreted as points in an n -dimensional space, it is possible to determine for a requested instance the K closest points out of the training set in terms of a distance metric e.g. Euclidean distance [79]. The most represented class among the k neighbors is expected to be the class of the requested instance.

(c) Decision tree

In the training phase decision tree algorithms recursively search for a proper attribute to partition the training data into more homogenous subsets [63]. The result is a hierarchical structure, where all internal nodes have an associated splitting attribute and all leaf nodes contain the related classes. A large amount of decision tree classifiers have been introduced in the literature; they mainly differ in the way to find the best attribute to split.

(d) Neural network

An artificial neural network is a simplified model of the brain consisting of interconnected nodes which simulate biological neurons. A threshold is associated with each node, a weight with each connection. If and only if the weighted sum of all inputs to a node exceeds its threshold the node fires [80]. In the training phase all thresholds and weights are adjusted until the outputs of the neural network for instances of the training set match their real classes.

(e) Support Vector Machine

SVM is a binary classifier that maps the input data into a sufficient-high dimensional space where the training instances become linearly separable. The separating hyperplane which maximizes the margin between it and the closest training instances is determined in the training phase and used for classification in the application phase.

2.4.3.1 Kernel Functions

SVM is a computing method based on statistical learning and optimization theories [16]. It is chosen as the classification algorithm in the recognition module because of its robustness in representing the information at the boundary class [17]. During the training process of SVM, it finds a set of hyperplanes to maximize the margin among themselves and the nearest data samples of arbitrary classes so that these hyperplanes are separable for data classification. SVM is initially designed to handle data of two classes where they are separated by

$$\mathbf{w}'\mathbf{x} + b = 0 \quad \text{Equation 2-1}$$

Where \mathbf{x} is the data sample, \mathbf{w} is the weight vector, and b is bias for constant offsets.

In many circumstances, a real-world data is complex. A linear SVM system may be not effective to separate this complex data that are non-linear. A way is to introduce a soft margin approach to handle non-linear problems. Another way to overcome this limitation of the SVM model is to include a non-linear kernel trick to make non-linear transformation of the data space to improve its recognition ability. In this case, the kernel tricks such as radial basis function, polynomial function and etc [18] can provide mapping from linear to non-linear classification.

2.4.3.2 SVM for Multiclass Classification Task

Agricultural track robot is essential to have the ability to classify more than two terrain types. SVM adopts two strategies to classify the data samples of multi-classes, i.e., either One-Versus-One (OVO) or One-Versus-All (OVA). The OVO strategy is firstly introduced in SVM [20] and it is also known as pairwise coupling or round robin. It is actually a basic form of binary classification. Let say n data pairs $D = \{x_m, y_m\}, m = 1, \dots, n$ are available for training, where $x_m \in \mathcal{R}^p$ is a feature vector indicating the m sample, and $y_m \in \{1, 2, \dots, K\}$ is the class label of x_m . The SVM model that implements OVO will consist of $K(K-1)/2$ binary SVMs. On the other hand, the OVA strategy is applied to build K SVMs where the i -th SVM is trained with all the data samples of the i -class coded as 1, and the data samples of other classes coded as -1. In this work, the SVM model is built to solve a problem by using an OVA strategy, as follows.

$$\text{Minimize} \quad \phi(\mathbf{w}_i, \boldsymbol{\xi}) = 0.5 * \|\mathbf{w}_i\|^2 + C \sum_{j=1}^{n_i} \xi_j^i \quad \text{Equation 2-2}$$

$$\begin{aligned} \text{Subject to} \quad & z_j (\langle \mathbf{w}_i, \phi(\mathbf{x}_j) \rangle + b_i) \geq 1 - \xi_j^i, \quad \text{sign}(z_j) = i \\ & z_j (\langle \mathbf{w}_i, \phi(\mathbf{x}_j) \rangle + b_i) \leq 1 - \xi_j^i, \quad \text{sign}(z_j) \neq i \\ & \xi_j^i \geq 0 \end{aligned} \quad \text{Equation 2-3}$$

where C is a predefined parameter being introduced according to a soft margin approach and it controls the trade-off between training accuracy and generalization (an example of the effect of C on a linear SVM is illustrated in Figure 5-4). The \mathbf{w}_j is the weight vectors of SVM trained with data samples from two classes; $\phi(\mathbf{x}_j)$ is the kernel function; b_i is a scalar; ξ_j^i is the slack variable that permits $i=1, \dots, n_i$ constraints to be violated; $z_j \in \{-1, 1\}$ is the class label for the classifier. Given a data sample \mathbf{x} , the decision function of the SVM is

$$\text{Class}(\mathbf{x}) = \arg \max_{i=1, \dots, K} (\langle \mathbf{w}_i, \mathbf{x} \rangle + b_i) \quad \text{Equation 2-4}$$

In higher dimensional feature space, kernel function is used to construct the mapping for the Support Vector Classification. Problem often arises when choosing the specific parameters (i.e. Kernels of features) process which affect the accuracy of the data are mapped.

The training data in two separate classes is defined as

$$D_T = \{(X_1, Y_1), \dots (X_i, Y_i), \quad X \in R^n, \quad Y \in \{-1, 1\}\}$$

Where

D_T is the training data, T is the sampling time

The hyperplane for mapping process

$$(W, X) + b = 0$$

Which said to be optimized if separated by the hyperplane is done without error with the distance between the vectors is maximized. With plotted region of

$$(W, X) + b \geq +1 \text{ if } Y_i = +1$$

$$(W, X) + b \leq -1 \text{ if } Y_i = -1$$

The Equation is the region that supports the hyperplane and no training data should be within the support plane. The training data should be outside or at the positive or negative support plane as shown in Figure 2.7.

FIGURE 2.7
Separating Hyperplane in the SVM between Two Data Sets

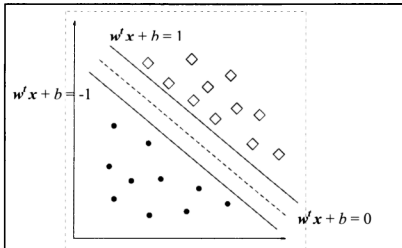
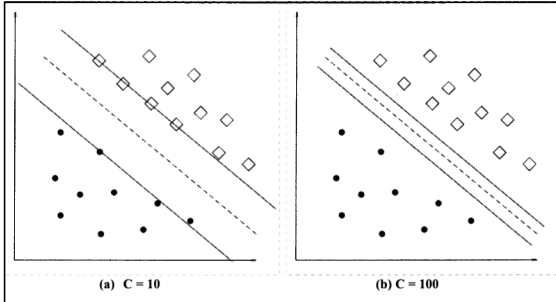


FIGURE 2.8

The effect of soft margin constant C . On the left side (a) $C = 10$ and at the right side (b) $C = 100$. The figure shows that the positive and negative samples can be separated by a hyperplane. In the case of (b), when the margin value increases, the hyperplane is closer to the boundary. By selecting an appropriate value of the parameter C , the SVM can perform with optimum classification results by reducing its training errors. [81]

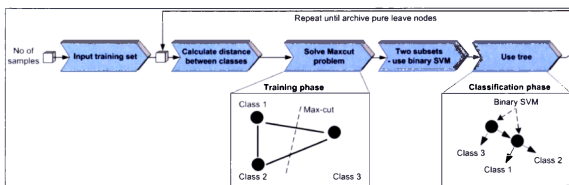


A numbers of kernel are deployed during the classification process. The kernels transform all the data set in a Euclidean Space where variation of methods used to during the classification. Popular kernels are use during the classification such as Polynomial, Quadratic, Linear and Radial Basic Function (RBF).

2.4.3.3 Hierarchical Support Vector Machine

Another technique to adopt SVM to multi-class problems is called Hierarchical Support Vector Machine [82]. The algorithm recursively partitions the set of classes into subsets. To determine a good split a max-cut problem is solved. Taking into account natural groupings using a distance measure the partitions with the maximum total distance between them are searched. Therefore HSVM promises not only high accuracy but also imposes little parameter tuning. Each internal node of the hierarchy represents a binary SVM. The partitioning of the data is stopped when all leaf nodes are pure containing only instances of one class shown in Figure 2.9.

FIGURE 2.9
Hierarchical Support Vector Machine



Each of the learning machines has its own tradeoff between accuracy and efficiency but in the terrain classification process, a robust and easy to use is the key in choosing the correct learning machine. Thus Support Vector Machine is selected as the main learning machine for the terrain classification. The Support Vector Machine also supports multiclass classification which is needed in classifying agriculture terrain.

CHAPTER THREE

DEVELOPMENT OF TRACK DRIVEN AGRICULTURE ROBOT

3.1 INTRODUCTION

This chapter aims to present on the development of the track robot mechanical structure design architecture. Within this chapter all the mechanical design such as the drive mechanism, flipper arm mechanism and the overall design is presented as well as the calculation, simulation and Computer Aided Design (CAD) used during prototype development. The track robot must be able to traverse across rough. The design paradigm introduced in this chapter to address such problem. The ideas proposed are

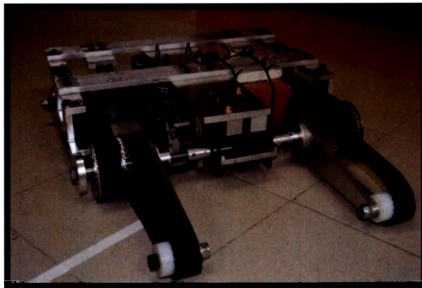
- a. The track robot is design with one DOF manipulator arm rather than only the track to provide better traction and stability during operation. The arm design must be strong enough to support its own weight.
- b. The center gravity of the track robot must be lower enough to maintain its stability during traversing on rough terrains or muddy terrains.

This chapter also covers the simulations on the track robot mechanical structure using ANSYS software which is performed to study the effect and expected capability of the design optimization. The designs are assembled and develop using CATIA 3D Computer Aided Design (CAD) software and then modeled on ANSYS software to perform the simulation analysis.

3.1.1 Mechanical Design

The track driven robot uses articulated tracks to improve traction on the agriculture field. The arm can be used in different mode of operations. The arm is used to provide better maneuverability and traction. The flipper arm systems also designed to enhance the ability to climb over an obstacle taller than the robot and shown in Figure 3.1.

FIGURE 3.1
Overall Mechanical Design of the Track driven robot



Both length of the robot are shown in Table 3.1 the mechanical specification for prototype development. Both dimension of the robot during extension and folding show in the table.

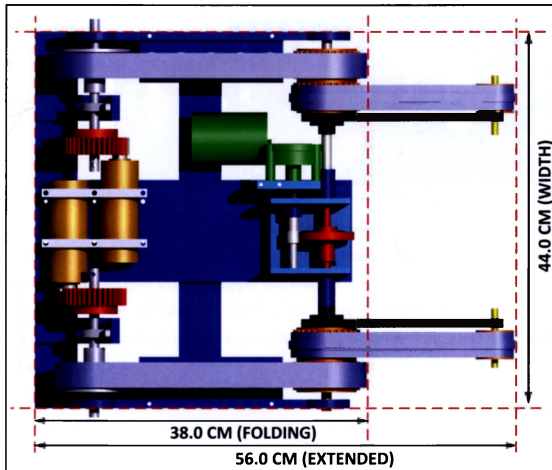
TABLE 3.1
The physical specification for the Track Robot

Overall weight of the robot	15 kg
Total length when the arms are fully extended	56.0 cm
Total length when the arms are fully folded	38.5 cm
Total width of the robot	44.0 cm

3.1.2 Motor Layout

The layout is divided into two parts, drive mechanism and flipper arm mechanism. The drive and flipper arm mechanism are carefully select based on data and performance analysis to ensure the motion is smooth and efficient. In the mechanical design, twin DC motor is situated at the back of the robot as the main propulsion system and single motor with worm gear module at the front is shown in Figure 3.2 shows the position for drive motor and flipper arm motor.

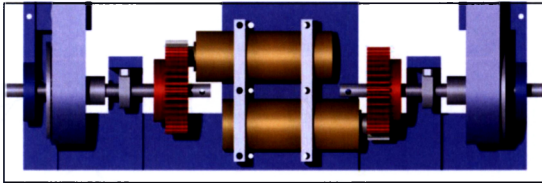
FIGURE 3.2
Position Drive Motor and Flipper Arm Motor



3.1.3 Drive Mechanism

The drive mechanism consists of two brushless DC motor propelling the robot. Driving mechanism is designed to be linked with passive wheel in front using a custom design articulated track system conveyed through 1:32.5 ratio planetary gearbox brushless dc motor. The motor are chosen due to the high efficiency and torque compared to normal gearbox system. The drive motor uses 24VDC with rated power up to 49.5 W, propelling the robot with 2.7 Nm continues torque.

FIGURE 3.3
Design architecture for the Drive Mechanism and gear ratios



The motion from the motor is transmitted through 1:3 ratio spur gears thus increasing the propelling torque to 8.1 N.m. The increase of the torque ratio is needed to compensate the load from the flipper arm module.

The motor selection is important to ensure a smooth motion during the operation. It is vital for the motor to provide enough torque and calculated using basic torque formula.

$$\tau = F \times d$$

Equation 3-1

Where

τ is the torque needed for the drive system

F is the force acting on the system

d is the distance between the force and the centre point of the torque

The equation is expanded to include the centre of gravity, CG , $d_{motor\ shaft}$ and number of drive motor, n . The equation is visualize in Figure 3.4 below

$$\tau = F \times (CG_{total} - d_{motor\ shaft}) \quad \text{Equation 3-2}$$

$$\tau = \frac{1}{n} mg \times \left(\frac{\sum (weight \times d_{CG})}{\sum weight} - d_{motor\ shaft} \right)$$

Where

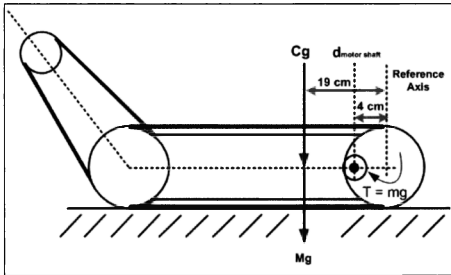
$d_{motor\ shaft}$ is the distance from shaft of motor to the reference axis

d_{CG} is the distance from centre of gravity to the reference axis

F is the mass times the gravitational acceleration

n is the number of drive motor

FIGURE 3.4
Diagram of Track-Driven Robot



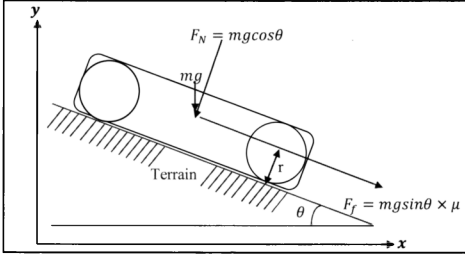
Substituting the parameters,

$$\tau = \frac{1}{2} (15) (9.81) \times \left(\frac{\sum (15 \times 0.19)}{\sum 15} - 0.04 \right)$$

$\tau = 2.7590 \text{ N.m}$ is needed for a single motor in normal operation

But in the real agriculture field, the worst case situation is assumed by introducing a ramp in the system. The calculation is derived from Equation 3-1, the force acting on the track robot divided into F_N and F_f elements illustrated in Figure 3.5.

FIGURE 3.5
Illustration of the worst case scenario



$$\tau = F \times d$$

Equation 3-2

$$\tau = (F_f + F_N) \times r$$

$$\tau = [(mg \sin \theta \times \mu) + (mg \cos \theta)] \times r$$

Where

θ = Angle of terrain

μ = Coefficient of static friction (Gravel)

The same case as the normal torque condition, the drive motor used in the system to drive the tracks is included in the calculations. Hence, the value of torque should be divided with the number of motor used to move the robot.

$$\tau = \frac{1}{n} [(mg \sin \theta \times \mu) + (mg \cos \theta)] \times r$$

Equation 3-3

Substituting the parameters, assuming the worse angle $\theta = 45^\circ$

$$\begin{aligned}\tau &= \frac{1}{n} [(mgsin\theta \times \mu) + (mgcos\theta)] \times r \\ \tau &= \frac{1}{2} [(15 \times 9.81 \times sin45 \times 0.85) + (15 \times 9.81 \times cos45)] \times 0.04 \\ \tau &= 3.8499 \text{ N.m}\end{aligned}$$

The calculation indicates that the maximum torque required in the worst case scenario the robot is 3.8499 N.m which more than the normal operation is 2.7590 N.m on a single track. In the design, two brushless planetary gears Dc Motor system is used with the specifications

- i. *Voltage (V) = 24V DC*
- ii. *Torque (τ) = 2.7916 Nm*

The default torque from the motor is not enough to support the robot during motion thus torque amplification is needed to support the requirement torque. A set of spur gear with ratio of 1:3 is used on each motor to amplify the torque. The new torque from the motor is calculated using

$$\begin{aligned}n_1\tau_{new} &= n_2\tau_{old} \\ (1)\tau_{new} &= (3)(2.7916) \\ \tau_{new} &= 8.3748Nm\end{aligned}$$

Where

- n_1, n_2 is the ratio of the mechanical system
- τ_{new} is the new torque required
- τ_{old} is the torque from the motor

The main disadvantage of using mechanical gear ratios is reduces in the efficiency. All loses in the system is due to the friction between gears producing heat and vibration thus reducing the gear performance. The comparison of efficiency between gears is listed in Table 3.2.

TABLE 3.2
Comparison between types of Gear

Gear Type	Shafts	Efficiency (%)
Spur	Parallel	80
Helical	Parallel/Perpendicular	80
Sprocket with Chain	Parallel	80
Bevel	Perpendicular	70
Rack and Pinion	-	90
Worm	Perpendicular	70
Planetary	Parallel	80

Based on the Table 4-2 spur gear has an efficiency of 80% and taking into account other factor such as teeth slippage. Thus:

$$\tau_{new} = 80 \% \times 8.3748 \text{ N.m}$$

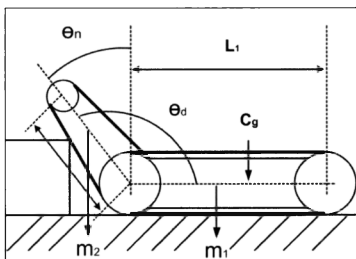
$$\tau_{80 \% \text{ new}} = 6.6994 \text{ N.m}$$

The calculation shows the reduction of efficiency of 20% producing torque of $\tau_{80 \% \text{ new}} = 6.6994 \text{ N.m}$ compared to the $\tau_{new} = 8.3748 \text{ N.m}$. The new torque calculation indicates the motor is suitable as the main drive system with extra torque than needed.

3.1.4 Flipper Arm Mechanism

For the flipper arm mechanism design, a single industrial brush motor is used to manipulate the flipper arm module. The motor is integrated with worm gearbox system which increase the torque tremendously but with drawback decreasing in speed. The worm gear easily turns the gear but not possible for the gear to do the opposite due to shallow angle generating friction holding the gears in to places. Flipper arm torque motor is calculated based on the worst case scenario of climbing obstacle which is higher than the robot itself and illustrated in Figure 3.6.

FIGURE 3.6
Agriculture Track Robot climbing obstacle.



Where

m_1	Mass of Robot
m_2	Mass of Robot Flipper
C_g	Centre of Gravity
L_1	Length between Front and Back Wheel
θ_n	Angle of the Flipper when contact with obstacle
θ_d	Angle of the Flipper to the main body

The same torque equation is used to calculate the required torque to support the entire track robot weight during operation.

$$\tau = F \times d$$

Equation 3-1

Where

τ is the torque needed for the flipper arm

F is the force acting on the system

d is the distance between the force and the centre point of the torque

Substituting all the values

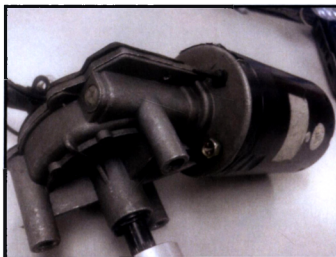
$$\tau = mg \times d$$

$$\tau = (15.0 \times 9.81) \times 0.2$$

$$\tau = 29.43 \text{ N.m}$$

Therefore, the required torque for lifting the 15 kg load is $\tau = 29.43 \text{ N.m}$. The flipper motors capable of delivering 8.0 N.m of torque based on the manufacturer datasheet which less than requirement and shown in Figure 3.7.

FIGURE 3.7
Automotive Dc Motor for the Flipper Arm



To compensate the lack of torque, an additional helical gear with 1:5 ratios is added for multiplying the torque to the flipper arm module. Using the similar formula

$$n_1 \tau_{new} = n_2 \tau_{old}$$

$$(1) \tau_{new} = (5)(8.0)$$

$$\tau_{new} = 40 \text{ N.m}$$

Where

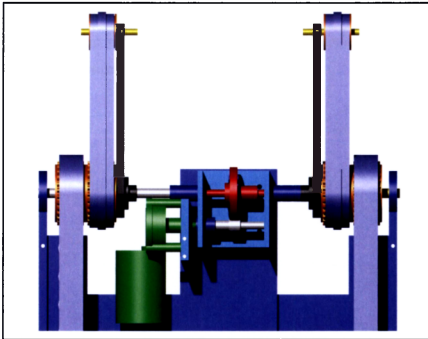
n_1, n_2	is the ratio of the helical gear
τ_{new}	is the new torque after the mechanical amplification
τ_{old}	is the torque from the industrial Dc Motor

Based on the Table 4-2, the helical gear has an efficiency of 80%. Thus:

$$\tau_{new} = 80 \% \times 40 \text{ N.m}$$

$\tau_{80 \% \text{ new}} = 32 \text{ N.m}$ Required for the robot and complete assembly is illustrated in Figure 3.8.

FIGURE 3.8
Complete Flipper Arm Assembly with Helical Gearbox.



3.2 MODELLING AND SIMULATION OF THE MECHANICAL STRUCTURE





3.2.1 Introduction

Simulations of the track robot mechanical structure were performed to study the effect and expected capability of the design optimization. The designs are assembled and develop by using CATIA 3D Computer Aided Design (CAD) software and then modeled on ANSYS software to perform the simulation analysis. Various interactions must be understood when designing the mechanical structure. ANSYS software is chosen as our simulation software due to its nature capabilities to virtually test the design and optimizing the mechanical design.

3.2.2 Design and Analysis

The designing process is done using Computer Aided Design (CAD) software CATIA and further analysis and modeling is done using Finite element analysis (FEA) in ANSYS software. During the analysis, all input data is multiply by five as the safety factor to guarantee the structure integrity during the actual testing. During the analysis process, the model is defined with ten nodes with three degree of freedom on each node. In the design, aluminum alloy is chosen as the construction material due to higher strength to weight ratio compared to steel. The material property for the alloy is observable in Figure 3.9.

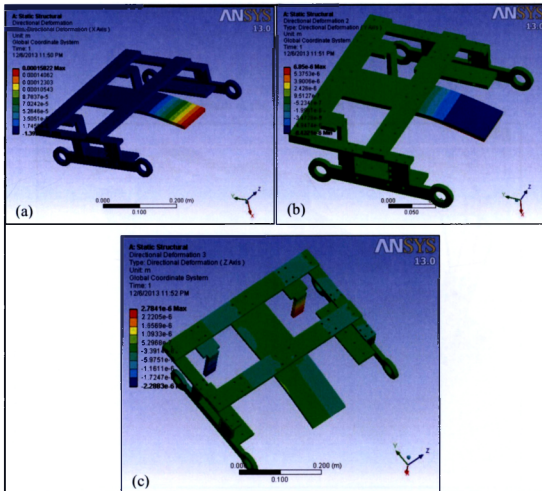
FIGURE 3.9
Properties of Aluminum Alloy

Properties of Outline Row 3: Aluminum Alloy NL			
	A	B	C
1	Property	Value	Unit
2	 Density	2770	kg m ⁻³
3	 Isotropic Elasticity		
4	Derive from	Young's Modulus ...	
5	Young's Modulus	7.1E+10	Pa
6	Poisson's Ratio	0.33	
7	Bulk Modulus	6.9608E+10	Pa
8	Shear Modulus	2.6692E+10	Pa
9	 Bilinear Isotropic Hardening		
12	 Specific Heat	875	J kg ⁻¹ C ⁻¹

3.2.3 Forces

The result from the analysis is observable in Figure 3.10 that the structure is able to withstand a force up to 200 N before the flipper arm motor holder fail to hold its integrity (in red color).

FIGURE 3.10
Directional Deformation X, Y, Z Axis based on the Applied Force: (a) X-axis, (b) Y-axis, (c) Z-axis.



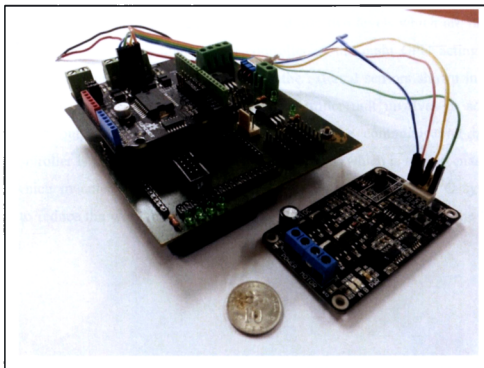
Based in Figure 3.10, the direction of deformation is applied to the X, Y and Z axis and producing different result on each axis. Also the back side did not show any critical deformation event with the amplified load applied to the hull structure (in blue color). Thus, the design is considered optimum because there is no sign of failing even after accumulated force is applied to the structure. The objective of total deformation is identifying changes in structure when force is applied to the structure.

3.3 DEVELOPMENT OF ELECTRONIC SYSTEM

3.3.1 Introduction

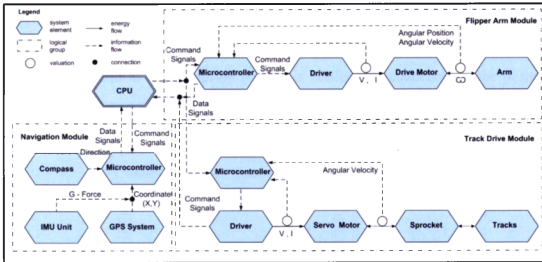
This chapter provides a detailed explanation of the electrical system which purposely developed for the agriculture track robot. Most of the electronics are custom designed to suite the requirement and maximizing the space and power usage for the robot platform shown in Figure 3.11. Internal space is not a luxury in the track robot and the design requirement needs it to be compact and efficient for maintenance and foolproof.

FIGURE 3.11
The Brain of the Track Robot System



The electronic system must be designed to be linked to the external element such as the analog input or output, digital input or output, and information processing. While many commercial 'off-the-shelf' products meet the robot requirement but the demand for custom PCB is required for the system to be integrated in a unique way because of the space and power requirement.

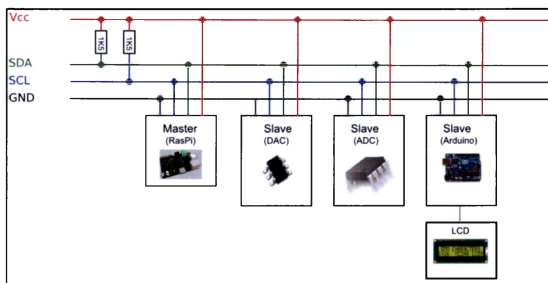
FIGURE 3.12
Electrical Design Architecture



The system design architecture is divided into two levels which are high and lower level controller. The higher level controller as the main CPU acting as the decision making unit based on the input from the external sensors shown in Figure 3.12. The lower level controller is the microcontroller unit involves in adjusting signals and output them to the actuating system. The communication between microcontroller is using an I²C or Inter-integrated Circuit which is a multi-master and slave which invented by Philips or NXP Semiconductor [83]. The I²C system is design to reduce the workload on the main CPU or master and distribute the load to slave units.

The I²C system transfers data using two wire which connected to bus through the SCL (serial clock line) and SDA (serial data line). Each devices connect thru the wire has a unique address and the device is recognizable based on the address. Any device which has the I²C capabilities can be connected and reducing the circuit space. For the I²C circuit to work, pull-up resistor is needed and connected to the positive supply of the system which shown in Figure 3.13.

FIGURE 3.13
Inter-integrated Circuit (I²C) for the Communication

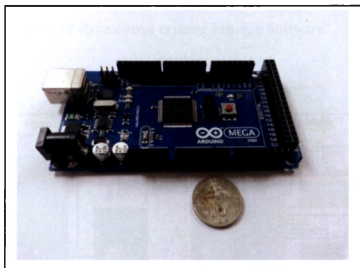


The master circuit always initiates the I²C SCL line and the slaves will respond to the master. The speed of the I²C is from 100 KHz and up to 3.4MHz is achievable but in the mobile robot only 100 KHz is used for data transferring. The maximum length for the I²C wire is about one meter and has a maximum 112 number of I²C devices.

3.3.2 Master Circuit

The board is the main primary CPU interface attached to the special enclosure in the robot ensuring no external element will affect the system.

FIGURE 3.14
Master circuit on the Agriculture Robot

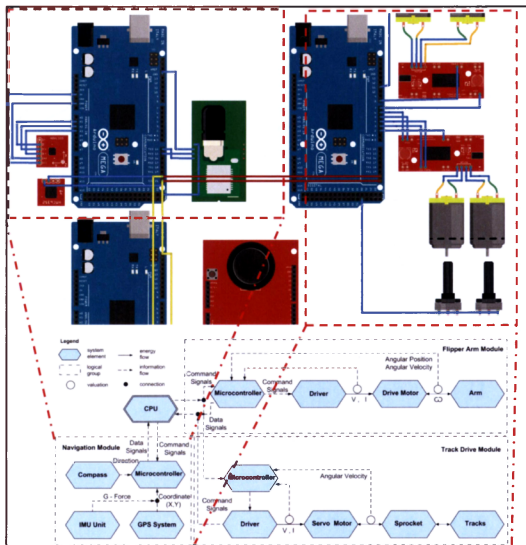


The master circuit shown in Figure 3.14 required special attention to ensure all external features are compatible with each other. Examples of the external features include measuring the analog/digital inputs, producing critical analog/digital output for the motors, decoding input values for the encoder and multi communication protocols. To perform such processing details, 8-bit Atmel AVR 2560 microcontroller was used. Two operating voltage 3.3VDC and 5VDC will be used for communication and I/O systems clocking over 16MHz. The system contained 54 digital pins for general input and output (I/O), and 16 analog inputs for reading the sensor with 10 bits of resolution with reference of 5 volts.

3.3.3 Slave Circuit

The master circuit will communicate with a slave microcontroller which is Atmel 8-bit AVR 128 for data and command signals using the I²C data line. The master and slave purpose is to reduce the load on the main CPU and enable faster processing speed and response. The slave circuit used to decode data from the sensors and pass it to the master circuit for decision making process. Figure 3.15 (Red color) shows the visualization of slave circuit is using Fritzing Software.

FIGURE 3.15
Slave Circuit using Fritzing Software



All the sensors such as Accelerometer, Compass and GPS are connector to the slave board for data processing thru I²C connection to the master circuit. For better understanding of the robot systems on its environment, sets of data must be extracted from the accelerometer and encoder then analyzed. The accelerometer used during the experiment is from SparkFun Electronics with triple axis accelerometer and was chosen because of fast data processing and easy to be integrated into the system.

3.3.4 Accelerometer

Accelerometer is based on MEMS (Micro- Electro Mechanical System) shown in Figure 3.16 that measure the physical acceleration experience by the object. The accelerometer measures in N/s^2 or g-force which allows instance force measurement relatively with the theory of proper acceleration. The motion of the sensors will be measured by two capacitors value that been held between two plates. The change in the capacitance value will be amplified producing voltage that is proportionate to the acceleration. When the sensor is placed on perfectly horizontal tables a value of 1g are measured upward to the vertical axis and 0g on the other axis. In engineering the G-force is quantified as acceleration or m/s^2 which is equal to $9.81 m/s^2$ of standard gravity.

FIGURE 3.16
Three Accelerometer from SparkFun Electronics

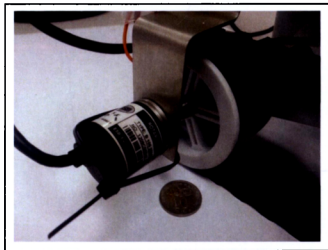


The sensor consume only 320uA thus making it extremely low in noise. The full sensing range is +/- 3G and data processing rate is up to 1600 Hz on X and Y axis and 550 Hz for the Z axis.

3.3.5 Encoders

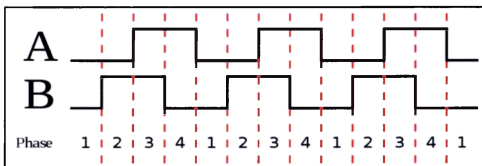
Encoder system is commonly used as an angular position feedback of a shaft. The encoder is an electro-mechanical system that converts the shaft motion into digital output and processed to gain information on speed, distance or position. There are two types of encoders generally used; they are called absolute encoder and incremental encoder shown in Figure 3.17. The absolute encoder works by preserving the position information even when the power supply is detached from the system and the information still accessible instantly when the power restored.

FIGURE 3.17
Encoder used during the Experiment



The incremental encoder working principle are different from the absolute encoder by providing pulses consist of A and B output square wave pulses shown in Figure 3.18 that will not preserve the last known information after a power cut off.

FIGURE 3.18
Quadrature phase of Encoder

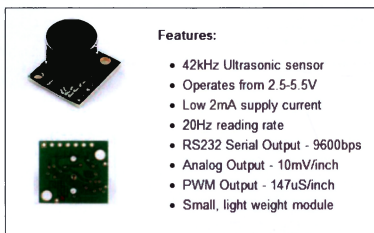


The output of 90 degree phase will be decoded by the microcontroller interrupt mode generating output and counting the direction or speed of the shaft. The encoder system will be connected to the track drive module to provide all the necessary data from the robot.

3.3.6 Ultrasonic-range finder (SN-LV-EZ1)

The Ultrasonic-range finder in Figure 3.19 used to determine the position of robot during the experiment. It detects an object at a distance without the needs for the robot to actually contact them. The sensor uses sound pulses to measure distance which has similarity with bats or submarines. The operation of this device is by emitting an ultrasonic pulse and timing how long it takes to hear an echo and can accurately estimate the distance between the object from its location. The ultrasonic-range finder can detect up to approximately 647 centimeters, but depending on the environment. Since the ultrasonic-range finder relies on sound waves, and best practice on any surface that deflect sound.


FIGURE 3.19
Ultrasonic-range finder (SN-LV-EZ1)



3.3.7 HMC6352 Compass module

The Compass HMC6352 module as shown in Figure 3.20 is used to keep the position of two ultrasonic-range finder pointing at north. Compass sensor is electrical navigational device that shows the directions of reference that is stationary relative to the surface of the earth. This compass HMC6352 is a fully integrated compass module that combines 2 –axis magneto-resistive sensors with the required analog and digital support circuits, and algorithms for heading computation. This compass module provides a simplest solution in terms of small size and cost effective in data collection.

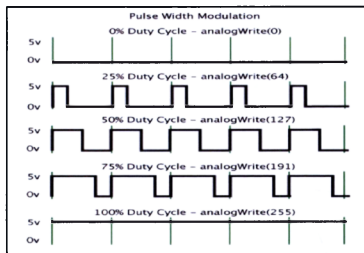
FIGURE 3.20
Magnetic compass (HMC6352)

	FEATURES <ul style="list-style-type: none">▶ Compass with Heading Output▶ Full Integration of 2-Axis Magnetic Sensors and Electronics▶ Firmware Included▶ Small Surface Mount Package (6.5 x 6.5 x 1.5mm, 24-pin LCC)▶ Low Voltage Operation (2.7 to 5.2V)	BENEFITS <ul style="list-style-type: none">▶ A Complete Compass. Everything is Done.▶ A Complete Digital Solution with Heading Output to Avoid Design of Hardware and Compassing Firmware Routines.▶ Data Acquisition, Calibration, and Heading Computation Routines Included for Quick-to-Market Designs.▶ Easy to Assemble & Compatible with High Speed SMT Assembly▶ Compatible for Battery Powered Applications
---	---	--

3.3.8 Driver

To control a DC Motors, a driver is needed to provide constant power and control signals to the actuators. The driver works when a control pulses and PWM are delivered from microcontroller allowing different speed and direction on the DC Motors. PWM or pulse width modulation is culprit in controlling the DC Motor speed or acceleration shown in Figure 3.21, the square wave signal works by means of switching voltage on and off rapidly by changing the portion time signals.

FIGURE 3.21
Duty cycle of Pulse Width Modulation (PWM)

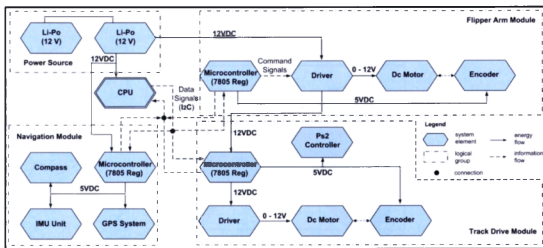


All the DC Motor uses the same driver to control the motion of the actuators. The drive is designed with capabilities to work with voltage 25VDC and the control frequency for the PWM is up to 20KHZ with solid state components to provide fast response eliminating mechanical wear.

3.3.9 Power Distribution

The robot will be powered by lithium polymer battery (LiPO) located inside the robot chassis. The power will be divided between the propulsion system, the main circuit board and sensors. The high power to weight/size ratios is the reason the battery was used allowing compact space within the robot.

FIGURE 3.22
Power Management Design



The source taken from the 12VDC LiPO is distributed along the microcontrollers and drivers shown in Figure 3.22. The microcontroller unit capable of operating in the range of 6 to 20VDC but recommended range by the manufacturer is between 7 to 12VDC to prevent overheated thus damaging the board. The build in 5VDC regulator on the microcontroller board is used to power up the microcontroller chip and harvested for powering sensors (such as in IMU or Compass) and ICs (such as POWER MOSFET in the Driver). The driver shares and extracts the same source from the microcontroller unit for driving Dc Motor which operates within the range of 0 to 12VDC. The microcontroller also has 3.3VDC regulator which is generated from the FTDI chip for running low voltage sensors and chips.

3.3.10 Communications

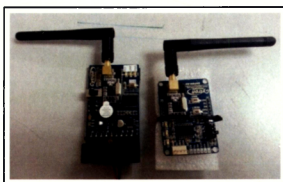
The robot can be either performs in manual or semi-autonomous operation. Semi-operation uses a wireless remote controller device allowing the user to interact with the robot using 2 joysticks and 14 digital buttons. The usage of joystick provides more flexible travelling direction for the robot with maximum range of 100 meter. The custom wireless systems are designed to be linked with Sony PlayStation controller in Figure 3.23 and direct implementation is achievable.

FIGURE 3.23
PlayStation Controller



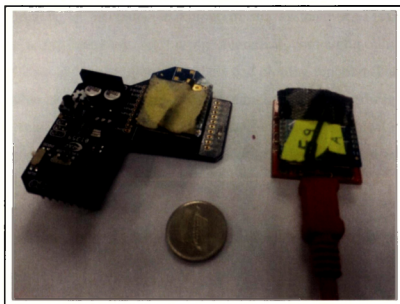
The controller was not specifically design for the robotic applications and a major adaptation and modification are needed to archive desired communications protocol. The remote uses 2.4GHz transceiver with frequency hopping technology for robust wireless connection. The system will automatically switch to another frequency band if interference is detected within the operating range.

FIGURE 3.24
Transmitter and Receiver Module for 2.4 GHz Wireless Systems



Another communication protocol used is the ZigBee. The ZigBee technology has been around for many years. It is popular among the user as a low cost and reliable solution for wireless feedback system networks. ZigBee product is intended for application with low energy requirement with constrained energy sources.

FIGURE 3.25
ZigBee Module: Transmitter and Receiver



During the experiment, set of data will be send through the ZigBee module for the classification algorithms. ZigBee consists of two parts – the transmitter and receiver module shown in Figure 3.25. The ZigBee protocol uses IEEE 802.15.4 radio communication standard. The system communication range is within 1.5 kilometers radius with 3.3V as the operating voltages.

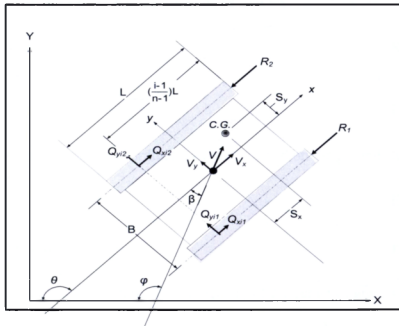
3.4 COORDINATE SYSTEM

Before the design and fabrication can be advanced, the kinematics for the track robot must be study and truly understood. The equation for the kinematics is based from M.Kitano and M.Kuma [1]. This chapter provides explanation on kinematics simulation of the track driven robot and compared to small scale track robot from actual system. The modeling for the track robot system was done using MATLAB Simulink to simulate the output of the system. MATLAB or Matrix Laboratory is a programming tool using numerical computing allowing matrix manipulations, data implementation algorithms and plotting function. The function on MATLAB enable in designing, implementing, simulating, image processing and etc. The Simulink function allows modeling the equation through graphical user interface for building models with assistance of a block library. Most of the kinematics models are based on the following assumptions

- a. Soil level is level
- b. The mass center is constantly on the midpoint of the track vehicle

The analysis on the track robot is done by isolating the frame in to X and Y reference frames illustrated in Figure 3.26. The coordinate is divided in two type of frame which defined as X-Y frame in fixed global and reference frame. The x-y frame is a local frame fixed at geometric center of the moving track vehicle. For all the time t , origin O stays at the center of gravity of the track robot.

FIGURE 3.26
Coordinate System for Tracked Vehicle Analysis.



Where

x	Direction of the heading robot in X axis
y	Direction of the heading robot in Y axis
θ	Yaw angle
V	Linear Velocity of the origin of moving axes
β	Side Slip Angle
ϕ	Directional Angle ($\theta - \beta$)
V_x	Velocity component in X axis
V_y	Velocity component in Y axis

The velocity V and forward and lateral accelerations a_x, a_y of the center of gravity are as follows:

$$\text{Velocity, } V = \sqrt{V_x^2 + V_y^2} \quad \text{Equation 3-2}$$

$$\text{Forward acceleration, } a_x = \dot{V}_x + V_y \dot{\theta} \quad \text{Equation 3-3}$$

$$\text{Lateral acceleration, } a_y = \dot{V}_y + V_x \dot{\theta} \quad \text{Equation 3-4}$$

Side slip angle β and side rate $\dot{\beta}$ are obtained as follows:

$$\text{Side slip angle, } \beta = \tan^{-1}\left(\frac{V_y}{V_x}\right) \quad \text{Equation 3-5}$$

$$\text{Side slip rate, } \dot{\beta} = \frac{V_x \dot{V}_y - V_y \dot{V}_x}{V^2} \quad \text{Equation 3-6}$$

The relations between φ and coordinates X_t and Y_t fixed on the ground shall be:

$$X_t = -\int_0^t V \cos \varphi dt \quad \text{Equation 3-7}$$

$$Y_t = \int_0^t V \sin \varphi dt \quad \text{Equation 3-8}$$

Therefore, the radius of curvature of the trajectory of the center of gravity shall be:

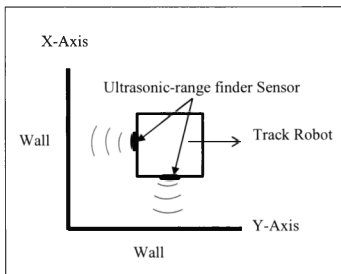
$$R_c = \frac{V}{\dot{\varphi}} \quad \text{Equation 3-9}$$

3.5 EXPERIMENTAL PROCEDURE

In this chapter, the tracked robot is used to compare the experimental result and simulation. In order to acquire the similar parameters as the simulation which is the input velocity and turning projector few sensors are used such as ultrasonic-range finder, encoder and compass. All of the sensors is connected to the Arduino Microcontroller and programming is loaded in to the microcontroller.

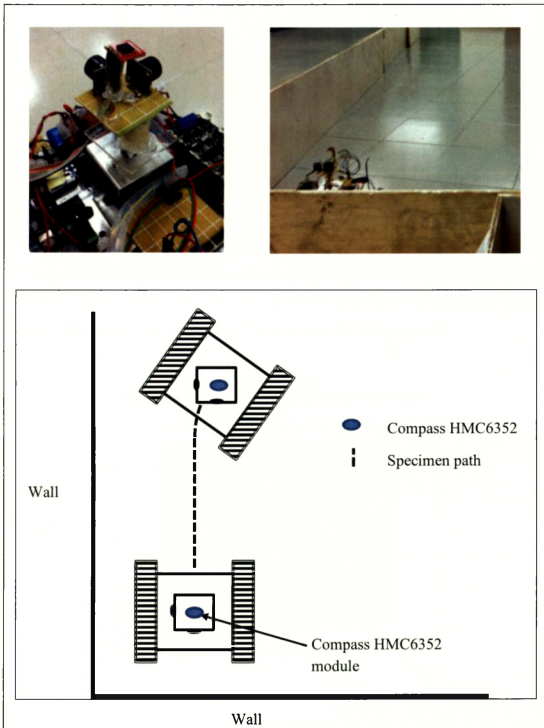
Two Ultrasonic Range Finder is used in this experiment where is used to detect distance in X-axis and Y-axis. Both ultrasonic-range finders are act as Cartesian plane and always at a fixed position when the track robot is moving. The sensors illustrated in Figure 3.27 when the data is collected.

FIGURE 3.27
The position of Ultrasonic-range finder during the experiment.



As discussed earlier, the compass shown in Figure 3.28 below is place on top of both the ultrasonic-range finder. The compass module is used to fix the position of the ultrasonic-range finder to north direction using a dc motor as an actuator to change the angle of the compass.

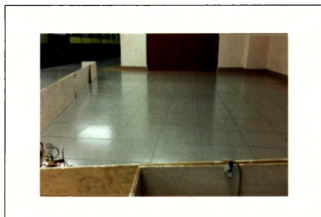
FIGURE 3.28
Position of both sensors during the experiment.



3.5.1 Experiment Process

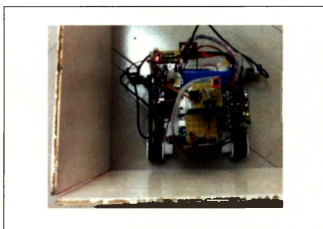
1. The experiment was done in spacious space with ceramic tiles floor as shown in Figure 3.29 in order for the specimen to move circularly as required.

FIGURE 3.29
Spacious space for experiment



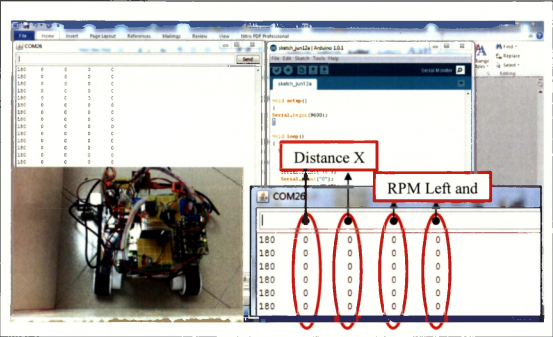
2. The wooden wall is assembled as shown in Figure 3.30 in order for the ultrasonic range finder to work out. The wall should be assembled 90° perpendicularly to each other so that the data of X and Y axis does not deviate greatly.

FIGURE 3.30
Wooden wall sets perpendicularly to the Track Robot



- As the wooden wall has been assembled, the specimen is place as shown in Figure 3.31. Then the specimen position is calibrated with data acquisition thru the wireless communication link. The specimen should be placed until the data of X and Y position 0 as shown in Figure 3.31.

FIGURE 3.31
Robot settings before the experiment starts.



- Then, the parameters required for trajectory motion is uploaded and programmed through USB cable. The parameters are shown in Table 3.3 below.

TABLE 3.3
The Parameter Input for Both Side Velocities.

No	Angular speed (RPM)	Velocity (ms^{-1})
1	170	0.534
2	113	0.356
3	85	0.267
4	57	0.178
5	29	0.090

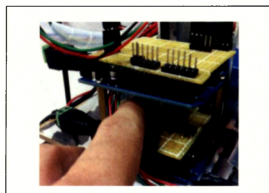
The parameters input as shown in Table 3.3 is required to be uploaded inside the Arduino microcontroller. There are 3 states need considered where, at state 1. Initially, both Left and Right track possess a same velocity of 0.534ms^{-1} , after 3 seconds. The right track speed will be reduce to the half of its original speed, 0.267ms^{-1}

TABLE 3.4
The Velocity Required for Each State at Specific Time

State	Time (s)	Linear velocity (ms^{-1})	
		LEFT	RIGHT
1	3	0.534	0.534
	22	0.534	0.267
2	3	0.356	0.356
	22	0.356	0.178
3	3	0.178	0.178
	22	0.178	0.090

- After all the input parameters been set, the code is then upload and ready to start the experiment.
- A switch button on the circuit board is pressed in order to start the motion of the robot as required according to the state, as shown in Figure 3.32.

FIGURE 3.32
The switch button is pressed to start the motion



7. As the robot is in mobile, the pathway is observed and recorded. After the robot stopped, the data transmit through serial monitor is recorded and investigate. The experiment is repeated for the next stage (2 and 3).
8. All the data acquire from the serial monitor is translated to graph in order to examine and investigate. The pattern between two graphs Simulink and experiment is discussed.

3.6 SIMULATION AND EXPERIMENTAL RESULTS

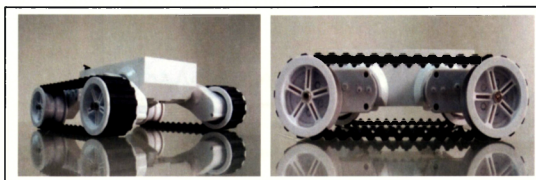
For the results, modeling is done using MATLAB to simulate the output of the system and compared to the small scale real model from the actual system. The result consists of simulated input track and trajectory motion of the robot at varying speed.

TABLE 3.5
List of parameter values of the Track Robot

PARAMETERS	VALUES
Vehicle thread, B	189 mm
Ground contact length, L	185 mm
Height of center of gravity CG, H	60 mm
Mass of vehicle, m	1893.7 g \approx 1.9 kg
Yawing moment of inertia, I_z	0.0111 kgm ²
Angle of approach and departure, θ_R ,	$\theta_R = 27^\circ$
θ_F	$\theta_F = 23^\circ$

In Table 3.5 shown all the parameters for the track robot used in the experiment. The results for both simulation and the experiment are discussed below.

FIGURE 3.33
Prototype of Small Scale Track Robot



3.6.1 Inner and Outer Velocities of the Specimen via Simulation

FIGURE 3.34
Inner and outer velocities of the robot via Simulation

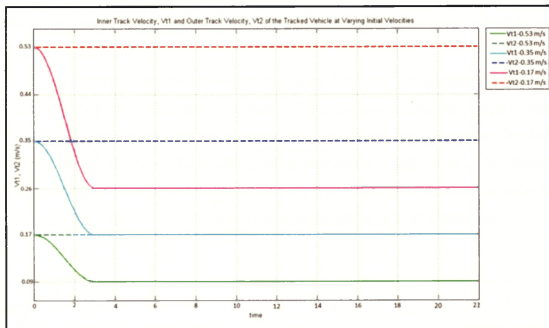
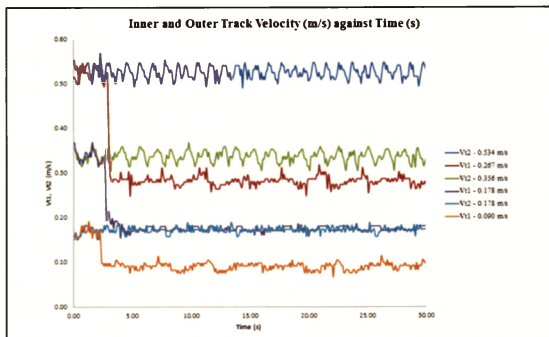


FIGURE 3.35
Inner and Outer Velocities of the Robot via Experimental



The analysis for simulation is assumed so that there are no losses for the sprocket-track linkage. From Figure 3.34 and Figure 3.35 shows the inner and outer velocity of the track via simulation and experiment. The experiment actually start when the robot start to move in circular motion. From Figure 3.34 and Figure 3.35, after it passes 3 seconds, the inner track velocity, V_{t1} is decelerate to half of its original velocity.

In experimental data as shown in Figure 3.35, the data acquire for relative velocity through the experiments are shown. The data possess fluctuation pattern due to the noise occur in the electronic circuit and mechanical flaw. However, the data acquire manage to shows the small variation similarities in pattern as in Simulation results in Figure 3.34. The experiment data for input velocity is essential in order to investigate the behavior of the trajectory motion for different kind of velocity.

3.6.2 Graph Trajectory Motion of the Tracked at Varying Forward Velocities for Simulation and Experimental

FIGURE 3.36
Trajectory of the tracked robot at, (a) 0.534 m/s via simulation, (b) 0.356 m/s via simulation and (c) 0.178 m/s via simulation

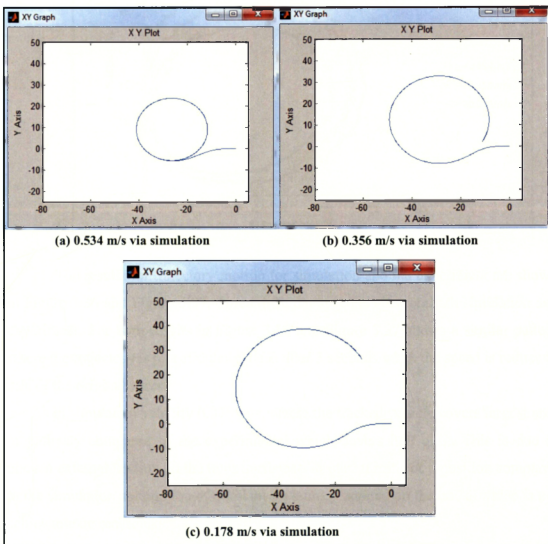
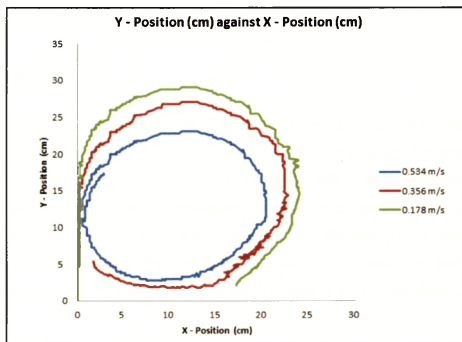


FIGURE 3.37
Trajectories of the Tracked Vehicle at Varying Initial Forward Velocities via Experiment



The results for trajectory motion for simulation and the experiment are shown in Figure 3.36 and Figure 3.37. Three different speeds is test on both simulation and experiment. For both results in Figure 3.36 and Figure 3.37 shows a similar pattern where the trajectory is in a circular pattern after 3 seconds when the speed is reduce to half of the original speed.

As in simulation velocity 0.178 m/s, where the tracked vehicle covers largest area of pathway compared to the experimental which covers half of it. This is due to uneven external friction on the track surface as its start turning at y-position compared to the simulation assumption of constant pressure is applied on the tracks which is not achievable on real tracks.

CHAPTER FOUR

TERRAIN CLASSIFICATION

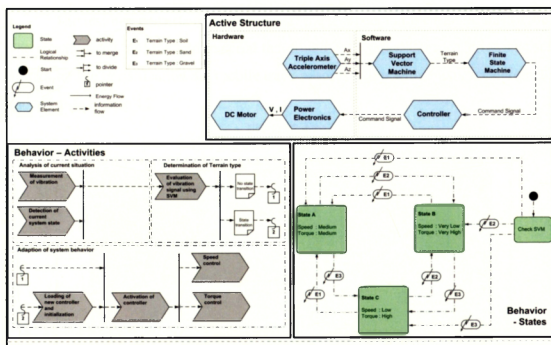
4.1 INTRODUCTION

Support Vector Machines (SVM) is a learning machine developed by Vapnik (1995) [84] and due to many fascinating features the system is gaining popularity between researchers. The algorithm is easy to use which offers good generalization performance and the algorithm has the flexibility to solve problem with little tuning on the system. The motivation when using Support Vector Machine is to produce an optimal separating hyper plane to distinguish between the data sets. The learning machine has advantages of being much more robust compared to other machines.

4.2 PRINCIPLE SOLUTION

The aim of this work is to evaluate the effectiveness of using Support Vector Machine (SVM) in recognizing terrain conditions in an agricultural field, i.e., sand, gravel and soil. For this purpose, a small tracked-driven mobile robot together with a terrain test bed has been developed. The key system elements of the agricultural robot developed for this work is shown in the active structure [85] at the upper part of Figure 4.1. The behavior state diagram and the behavior state activities are shown at the lower part of Figure 4.1.

FIGURE 4.1
Principle Solution of a Track Agriculture Robot with Terrain Classification Functionality



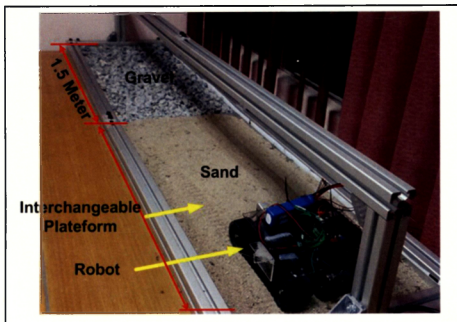
4.3 DEVELOPMENT OF THE TEST BED

In this chapter, the development of test bed setup is discussed. The motivation for the project is to control the track robot motion speed based on the terrain classified by a classifying algorithms. The mobility of the track robot is important in the robot overall performance. The objective for this chapter is to design and integrating the test bed with the track robot to creating a control environment for the system. The result data from the test bed were used to classify using Support Vector Machine to distinguish between terrains.

4.3.1 Experimental Setup

The test bed consists of two modules: the test bed and the mobile track robot. The terrains used are standard material that found in the area. In order to retrieve sets of data for the classification the mobile track robot is equip with 3 axis accelerometer and setup as Figure 4.2.

FIGURE 4.2
Test Bed Setup and Mobile Robot



An interchangeable terrain is placed in the test bed with a range of 1500mm on each terrain. The facility contains a flat rectangular size 1500mm by 500mm and filled up to 50mm depth. The mobile robot is positioned at the end of the test bed as the starting position. A three type of interchangeable terrain are sand, gravel and soil is used during the experiment. These terrains are distinguished based on the grain size shown in Figure 4.3. In the experiment, the grain size for the soil ranges between 1 to 20mm, the grain size for the sand ranges between 0.3 to 3mm while the gravel has a grain size of 5 to 35mm. The variation of grain size results in different vibration profiles during the experiment.

FIGURE 4.3
Different type of terrains (a) Gravel, (b) Sand and (c) Soil

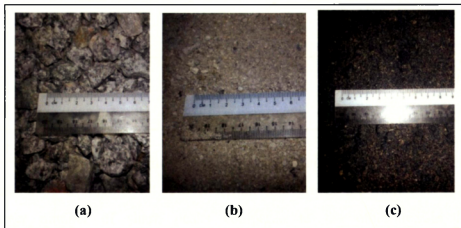
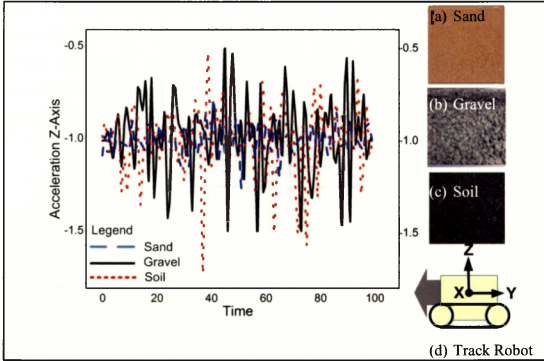


FIGURE 4.4

Example of data taken using IMU on a different type of terrain; (a) Sand, (b) Gravel, (c) Soil, (d) Robot

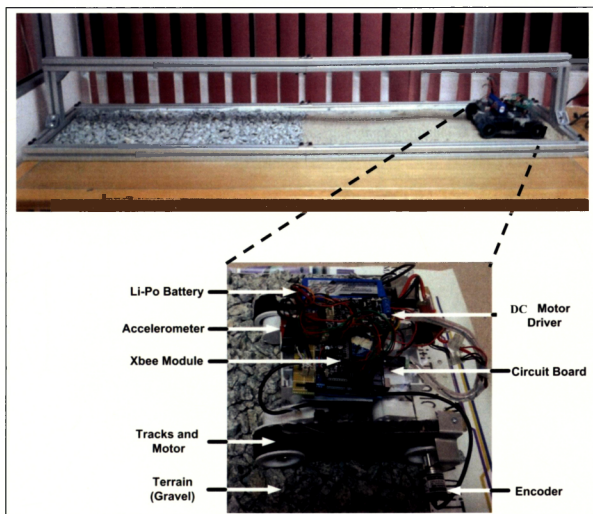


The vibration profile plotted a significant difference between terrains is observable in Figure 4.4. For a terrain like gravel the vibration is producing a larger number of sharp peaks compared to the other terrain like sand experienced less fluctuating nature with smaller peaks. The data then is split and rearrange in to a vector and time (ms) for classification process later.

4.3.2 Mobile Track Robot

A small scale mobile track robot is used during the experiment, creating a control environment for data collection. The track robot consists of two track drive to propel the robot forward or reverse. An accelerometer from SPARKFUN Electronics is used and capable to measure acceleration up to $\pm 3G$. A microcontroller unit is located at the top of the track robot for executing driving motion and data acquisition process.

FIGURE 4.5
Testing and Data Acquisition



The data acquired by the microcontroller is send via wireless XBee transmitted to the nearby laptop computer for data logging. The IMU is directly mounted on the mobile robot body without a suspension system which will absorb and reduce the accuracy of data reading.

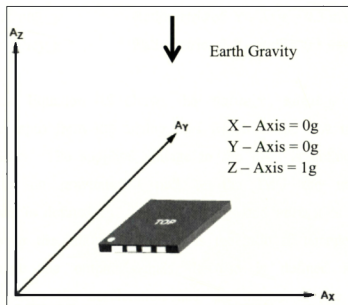
For data collection the robot is placed at the designated starting zone of the test rig comprises gravel. The test rig is designed with aluminum frame on the sides which cater for the robot to move in a straight line. After the Li-Po battery is connected and switched on, a warm start for around 15 seconds is needed for the Xbee Transmitter to function properly. The Xbee Receiver is connected to the USB data server and the port is configured to the designated communication port. The measured data (acceleration along the X, Y and Z axes and the speed of the left and right motors) will be transmitted continuously to the data server until the desired data sets have been achieved. The procedure is repeated for sand and soil.

For each terrain type data is measured for both acceleration in X, Y, Z axis and speed with 75% PWM. Then the measurement is repeated with different speed of 25% and 50% PWM level respectively.

4.3.3 Calibration Process

Before an accurate value of accelerometer can be collected, a manual calibration process should be performed. The IMU actually has been calibrated as a factory default, but manual calibration should be performed to ensure continues flow of accurate measurement data is obtained. The IMU is able to measure range of acceleration between (- 3 g to + 3g) with the supply range of (1.8 V to 3.6 V). The three outputs which is the X-axis, Y-axis and Z- axis is used during the data collection. The IMU not only able to measure the acceleration but also the orientation which is the force pointing to the center of the earth. It is also should be noted that calibration process of the IMU will affect the system performance. Before the calibration process started, the IMU is placed on the top of flat surface shown in Figure 4.6 below.

FIGURE 4.6
IMU position during the calibration process



During this process the ideal value for the X-axis is 0 g, Y-axis is 0 g and Z-axis should be 1 g which pointing to the center of the earth. The raw value of the accelerometer is observed using IDE of ARDUINO which feed the value to the serial communication online. The raw value is calculated using formula below to obtain the same value as Figure 4.6.

$$X_{axis} = (value_x/1024.0 * ADC_ref - zero_x)/sensitivity_x; \quad \text{Equation 4-1}$$

$$Y_{axis} = (value_y/1024.0 * ADC_ref - zero_y)/sensitivity_y; \quad \text{Equation 4-1}$$

$$Z_{axis} = (value_z/1024.0 * ADC_ref - zero_z)/sensitivity_z; \quad \text{Equation 4-1}$$

where

<i>value_x</i>	Raw value from X - Axis
<i>value_y</i>	Raw value from Y - Axis
<i>value_z</i>	Raw value from Z - Axis
<i>ADC_ref</i>	Supply voltage to the IMU (3.3V)
<i>zero_x</i>	Voltage when the X - Axis = 0g (1.65V)
<i>zero_y</i>	Voltage when the Y - Axis = 0g (1.65V)
<i>zero_z</i>	Voltage when the Z - Axis = 0g (1.65V)
<i>sensitivity_x</i>	Ratiometric of X - Axis = 0.3 mV/g
<i>sensitivity_y</i>	Ratiometric of Y - Axis = 0.3 mV/g
<i>sensitivity_z</i>	Ratiometric of Z - Axis = 0.3 mV/g

From the Equation 4.1 above, the *value_x*, *value_y* and *value_z* is defined as the input from the analog pins which is the raw input value from the accelerometer. The supplied voltage to the IMU is defined as *ADC_ref*. When there is no gravitational pull to the IMU the voltage on each respectively axis is defined as *zero_x*, *zero_y* and *zero_z* is equal to 1.65V. The sensitivity or the ratiometric output is the ratio of change in acceleration when to change the output signals. It also is defined as straight line relationship between the acceleration and the output of the IMU where is equal to 0.3 mV/g. When all of the value is tuned, the Equation 4.1 is uploaded into the Arduino for data collection process.

CHAPTER FIVE

RESULTS AND ANALYSIS

5.1 INTRODUCTION

This chapter is the last chapter, presenting the data and analysis on the Support Vector Machine (SVM) and Hierarchical Support Vector Machine (HSVM) method for terrain classifications. The data comprising three types of terrains such as sand, gravel and soil. During the classification process, four type of kernel function is deployed separating terrain into category which is polynomial, quadratic, linear function and radial basis function.

By measuring the acceleration values over time, a characteristic signal for each terrain type is produced. Sand is characterized by low amplitude of vertical acceleration. For gravel and soil higher amplitudes are observed. A difference between gravel and soil is the impact on the track robot speed. During the classification process there are few things are investigated using the following attributes for terrain classification:

- a) Standard deviation of acceleration in Z-axis
- b) Mean of current speed

For improvement of the results, a new feature is introduced during the classification process which is to consider the acceleration in other directions. In the second experiment the Support Vector Machine is trained using the following features:

- a) Standard deviation of acceleration in X-axis
- b) Standard deviation of acceleration in Y-axis

5.2 RESULT AND DISCUSSION

The aim of the experiments is to make statements about the efficiency of different SVM variations in classifying agricultural terrains, i.e. gravel, sand and soil. By comparing the One-Versus-All strategy using four different kernel functions and the HSVM using two different distance measures shown in Table 5.1.

TABLE 5.1
Accuracy of Classification using Speed Information and Acceleration in Z Axis

Classifier	Kernels /Distance	Gravel	Sand	Soil	Mean
SVM	Linear Kernel	99.33%	81.67%	63.44%	81.48%
	Quadratic Kernel	99.33%	84.67%	85.44%	89.81%
	Radial Basis Kernel	99.33%	84.67%	85.44%	89.81%
	Polynomial Kernel	99.33%	84.67%	85.44%	89.81%
HSVM	Pseudo KL Distance	99.00%	63.67%	89.00%	83.89%
	Euclidean Distance	99.00%	63.67%	89.00%	83.89%

TABLE 5.2
Accuracy of Classification using Speed Information and Acceleration in X, Y and Z Axis

Classifier	Kernels /Distance	Gravel	Sand	Soil	Mean
SVM	Linear Kernel	97.22%	91.44%	75.00%	87.99%
	Quadratic Kernel	98.44%	91.44%	90.78%	93.55%
	Radial Basis Kernel	98.44%	91.44%	90.78%	93.55%
	Polynomial Kernel	98.44%	91.44%	90.78%	93.55%
HSVM	Pseudo KL Distance	98.67%	88.33%	87.00%	91.33%
	Euclidean Distance	98.67%	88.33%	87.00%	91.33%

To make a comparison between the different variations the accuracy measurement is used by identifying the percentage of correctly classified instances among all classified. During the training phase, the classifier with nine parts from data is used and the remaining part for measuring the accuracy. The process is repeated 10 times using different instances for testing each time and the accuracy was averaged over the 10 folds. The advantage of this 10-fold cross validation is that during the process all samples are used for training and testing as well.

Only the vertical vibration or Z axis is used to identify the terrain type. In the first experiment and acceleration in X and Y axis is excluded from the training process. Terrain (Gravel) is recognized with high accuracies for SVM and HSVM variations shown in Figure 5.1. For the SVM for Terrain (Sand, Soil), Linear kernel shown a lower accuracy compared to Quadratic, RBF and Polynomial. This is due to the data plotted on the Euclidean plane cannot be effectively separated by Linear kernel. In the average, the quadratic, radial basis and polynomial kernel functions perform the highest mean of 89.81% and for HSVM kernels shown a mean of 83.89%.

The second experiment is done to investigate whether the classification performance can be improved by incorporating the acceleration in X and Y axis and shown in Table 5.2. For Linear function the terrain (Gravel, Sand) shows a high accuracy but reduce in accuracy for terrain (Soil). For other kernels, shows a higher accuracy of 93.55% compared to the Linear function only 75%. The HSVM kernel also outperforms the linear function with mean of 91.33%.

A kernel trick is used in the classification due to the nature of the data and to maximize the separating line in the hyperplane. There is no kernel that can fit all the data nature which can be linear and non-linear or both. Kernels function also can be interpreted as measuring a similarity between the input objects. There is no specific way to determine the best kernel for data sets. An experiment is needed with different kernel and model parameter is adjusted to minimize the error. In the result shown that most of the linear function kernel has the same accuracy for gravel and lower classification rate for the sand and soil.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 CONCLUSION

The primary objective of this research is to find a way to develop a track-driven agriculture robot with terrain classification functionality and has been archived. The aim is separated in to three sections, track robot development, terrain classification, simulation and experimental study

The track robot development began with the development of mechanical and electrical system of the robot. The mechanical design is fabricated based on the problem occurs, as discussed in the development chapter. The custom electronic design allowed the robot to pass all the information thru the main processor and wireless system allowed the robot to be tested in the test bed without being tethered. The electrical system were integrated into the mechanical design and maximizing the usage of the space in the robot chassis design. Sensors such as accelerometer were successfully providing the information to the robot which helps in the classification process.

In order to understand the robot navigation process, a mathematical model were develop and using MATLAB to simulate the trajectory path of the track robot. To prove the simulation process, an experiment was done using a small scale track robot on a different speed. The ideal trajectory of the robot was found out using both simulation and experiment and compared. It was found out that the trajectory of the robot has similar pattern on both simulation and experiment thus proving the simulation models.

To develop a terrain classification robot with realizing the functionality of terrain classification, a learning machine is needed and one of the possible learning machines is the Support Vector Machine (SVM). The vibration signals as a result of track-terrain interaction have been measured and used along with information about the speed as an input for two different SVM types, i.e. SVM using One-versus-All-strategy and HSVM. The

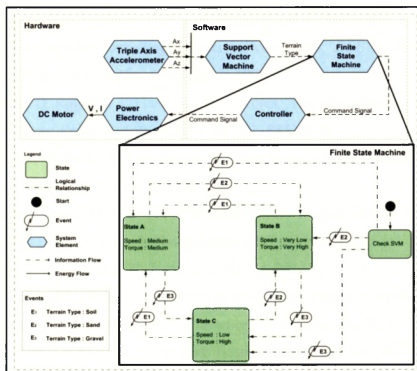
effectiveness of the SVM using four kernel functions, i.e. linear, quadratic, radial basis and polynomial function, as well as the HSVM using pseudo KL and Euclidean distance have been compared by applying the 10-fold cross validation method. Although the training set contains data recorded at different speed levels new instances can be classified with high accuracy. Classification performance can be improved considering not only vertical vibration, but also acceleration in X-Axis and Y-Axis. The results show that the SVM using quadratic, radial basis or polynomial kernel fulfill the task of terrain classification best. In conclusion, the finding in this research was to develop a track driven agriculture robot has been established.

6.2 RECOMMENDATION

The Support Vector Machine and Hierarchical Support vector Machine were used in the experiment in MATLAB environment. The current progress is an off-line classification method. The data is manually collected from the robot using wireless communication and the Support vector Machine is executed on a different computer. It is recommended that the Support Vector Machine is developed in Labview environment and integrated in the robot for real time classification. It is also suggested to develop a visual technique and integrate it with the current system to increase the efficiency of the system in classifying terrains.

Integrated Finite State Machine in next generation terrain classification robot in Figure 6.1 presents the system architecture for future terrain recognition in this research work. The system consists of two components; hardware architecture and software algorithm. A track driven mobile robot is the hardware in this application. The track robot consists of two tracks which maneuvers using differential drive system. In the software architecture, the system was developed under the MATLAB environment and C programming language. Current SVM only classify the terrain without any decision making algorithm for controller optimization.

FIGURE 6.1
Future Improvement for Terrain Classification



The terrain classification is composed by an SVM with an output that represents one of the three terrain types. Then the Finite State Machine (FSM) is applied to suggest a motor configuration based on the state triggered in the event for example event, E1 for soil will lead to normal motor speed and torque or E2 for sand will trigger very low speed and very low torque configuration or E3 for gravel will trigger the configuration with low speed and high torque. The decision making made by the Finite State Machine is used by the controller sending command to power electronics for motor drive motion based with different speed and torque. By providing such feedback to the controller, the robot manages to prevent itself from being trapped when traversing on different types of terrain.

REFERENCES

- [1] B. Allotta, G. Buttazzo, P. Dario, and F. Quaglia, "A force/torque sensor-based technique for robot harvesting of fruits and vegetables," *EEE Int. Work. Intell. Robot. Syst. Towar. a New Front. Appl.*, 1990.
- [2] Y. Edan, D. Rogozin, T. Flash, and G. E. Miles, "Robotic melon harvesting," *IEEE Trans. Robot. Autom.*, vol. 16, pp. 831–835, 2000.
- [3] F. Sistler, "Robotics and intelligent machines in agriculture," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 3–6, Feb. 1987.
- [4] C. Low, N. Aziz, M. Aldemir, R. Dumitrescu, H. Anacker, and M. Mellado, "Strategy planning for collaborative humanoid soccer robots based on principle solution," *Prod. Eng.*, vol. 7, no. 1, pp. 23–34, 2013.
- [5] H. Mousazadeh, "A technical review on navigation systems of agricultural autonomous off-road vehicles," *Journal of Terramechanics*, 2013.
- [6] S. Lee and D. Kwak, "A terrain classification method for UGV autonomous navigation based on SURF," *2011 8th Int. Conf. Ubiquitous Robot. Ambient Intell.*, pp. 303–306, Nov. 2011.
- [7] K. H. Low, G. J. Gordon, J. M. Dolan, and P. Khosla, "Adaptive Sampling for Multi-Robot Wide-Area Exploration," *2007 IEEE Int. Conf. Robot. Autom. Roma, Italy*, no. April, pp. 10–14, 2007.
- [8] S. a. Hiremath, G. W. a. M. van der Heijden, F. K. van Evert, A. Stein, and C. J. F. ter Braak, "Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter," *Comput. Electron. Agric.*, vol. 100, pp. 41–50, Jan. 2014.
- [9] G. Belforte, R. Deboli, P. Gay, and A. Giglio, "Robot design for applications in intensive agriculture," *2002 IEEE Int. Conf. Ind. Technol. 2002. IEEE ICIT '02.*, vol. 1, pp. 519–523, 2002.
- [10] C. Zhao, G. Jiang, and E. Engineering, "Baseline Detection And Matching To Vision-Based Navigation Of Agricultural Robot," no. July, pp. 11–14, 2010.
- [11] W. Ji, D. Zhao, F. Cheng, B. Xu, Y. Zhang, and J. Wang, "Automatic recognition vision system guided for apple harvesting robot," *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1186–1195, Sep. 2012.

- [12] M. Sciences and S. Alam, "Assessment of palm oil fresh fruit bunches using photogrammetric grading system," vol. 18, no. 3, pp. 999–1005, 2011.
- [13] Y. Huang, Y. Lan, S. J. Thomson, A. Fang, W. C. Hoffmann, and R. E. Lacey, "Development of soft computing and applications in agricultural and biological engineering," *Comput. Electron. Agric.*, vol. 71, no. 2, pp. 107–127, May 2010.
- [14] J. Wang, D. Zhao, W. Ji, J. Tu, and Y. Zhang, "Application of support vector machine to apple recognition using in apple harvesting robot," *2009 Int. Conf. Inf. Autom.*, pp. 1110–1115, Jun. 2009.
- [15] A. Baerveldt, "An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control," pp. 21–35, 2002.
- [16] A. Shukla and S. S. Jibhakate, "Design and implementation of real time pollution free autonomous vehicle for harvesting on VI platform," *2011 3rd Int. Conf. Electron. Comput. Technol.*, pp. 335–339, Apr. 2011.
- [17] F. Qingchun, Z. Wengang, Q. Quan, J. Kai, and G. Rui, "Study on Strawberry Robotic Harvesting System," pp. 320 – 324, 2012.
- [18] B. Yoon, "Design of paddy weeding robot," *Ieee Isr 2013*, pp. 1–2, Oct. 2013.
- [19] S. M. Abbas and A. Muhammad, "Improvements in Accuracy of Single Camera Terrain Classification," 2012.
- [20] C. J., M. Krishna, and C. V. Jawahar, "Fast and Spatially-Smooth Terrain Classification Using Monocular Camera," *2010 20th Int. Conf. Pattern Recognit.*, pp. 4060–4063, Aug. 2010.
- [21] D. Song, C. Yi, I. H. Suh, and B.-U. Choi, "Self-supervised terrain classification based on moving objects using monocular camera," *2011 IEEE Int. Conf. Robot. Biomimetics*, pp. 527–533, Dec. 2011.
- [22] C. Brooks, K. Iagnemma, and S. Dubowsky, "Vibration-based Terrain Analysis for Mobile Robots," no. April, pp. 15–20, 2005.
- [23] M. Suzuki, E. Terada, T. Saitoh, and Y. Kuroda, "Vision Based Far-Range Perception and Traversability Analysis using Predictive Probability of Terrain Classification," *ISR / Robot. 2010 Vis.*, pp. 50–55, 2010.
- [24] F. L. Garcia Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel, and R. S. Fearing, "Performance analysis and terrain classification for a legged robot over rough terrain," *2012 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 513–519, Oct. 2012.

- [25] S. Krivi, A. Mrzi, and N. Osmi, "Building Mobile Robot and Creating Applications for 2D Map Building and Trajectory Control," *MIPRO 2011*, pp. 1712–1717, 2011.
- [26] H. Jian-hai, Z. Shu-shang, L. Ji-shun, and L. Hang, "Research on Developed Parallel Two-wheeled Robot and Its Control System," no. September, pp. 2471–2475, 2008.
- [27] C. Ye, "Kinematic Analysis on a Mobile Robot Composed of Three Wheeled Units," pp. 485–489, 2008.
- [28] M. Lauria, I. Nadeau, P. Lepage, Y. Morin, P. Giguere, D. Letourneau, F. Michaud, and F. Gagnon, "Design and Control of a Four Steered Wheeled Mobile Robot," vol. 00, pp. 0–5, 2002.
- [29] V. Nazari and A. K. Modeling, "A Vision-based Intelligent Path Following Control of a Four-wheel Differentially Driven Skid Steer Mobile Robot," no. December, pp. 17–20, 2008.
- [30] T. Yoshihiro, "Development of a Wheeled Mobile Robot ' Octal Wheel ' Realized Climbing up and Down Stairs," pp. 2440–2445, 2004.
- [31] A. Mandow, J. L. Mart, A. Garc, and J. Gonz, "Experimental kinematics for wheeled skid-steer mobile robots," pp. 1222–1227, 2007.
- [32] F. Arvin, K. Samsudin, and M. A. Nasser, "Design of a Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation," no. July, pp. 143–147, 2009.
- [33] S. Shair, J. H. Chandler, V. J. González-Villela, R. M. Parkin, and M. R. Jackson, "The Use of Aerial Images and GPS for Mobile Robot Waypoint Navigation," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 6, pp. 692–699, Dec. 2008.
- [34] P. Features, "Pioneer 3-DX."
- [35] J. L. Mart, A. Mandow, A. Peque, and A. Garc, "Simplified Power Consumption Modeling and Identification for Wheeled Skid-Steer Robotic Vehicles on Hard Horizontal Ground," pp. 4769–4774, 2010.
- [36] Z. Li, S. Ma, B. Li, M. Wang, and Y. Wang, "Design and basic experiments of a transformable wheel-track robot with self-adaptive mobile mechanism," *2010 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 1334–1339, Oct. 2010.

- [37] G. Kim, S. Kim, and Y. Hong, "A robot platform for unmanned weeding in a paddy field using sensor fusion," ... *(Case), 2012 IEEE ...*, pp. 1–4, 2012.
- [38] X. Duan, Q. Huang, N. Rahman, J. Li, and J. Li, "MOBIT , A Small Wheel - Track- Leg Mobile Robot *," pp. 9159–9163, 2006.
- [39] A. C.-C. B. T. Coal and M. Exploring, "Effects of the Fiber Releasing on Step-climbing Performance of the Articulated Tracks Robots," pp. 818–823, 2009.
- [40] F. Michaud, D. Lctoumeau, M. Arsenault, Y. Bergeron, and R. Cadrin, "AZIMUT , a Leg-Track-Wheel Robot," no. October, pp. 2553–2558, 2003.
- [41] F. Patané, V. Mattoli, C. Laschi, B. Mazzolai, P. Dario, S. Superiore, and S. Anna, "Biomechatronic Design and Development of a Legged Rat Robot," pp. 847–852, 2008.
- [42] N. Fujii and K. Ohnishi, "Smooth Transition Method from Compliance Control to Position Control for One Legged Hopping Robot," pp. 164–169, 2006.
- [43] G. S and B. Luk, "Evolutionary Design And Development Techniques For An 8-Legged Robot," *Genet. Algorithms Eng. Syst. Innov. Appl.*, no. 446, pp. 2–4, 1997.
- [44] M. Hutter, C. D. Remy, M. A. Hoepflinger, and R. Siegwart, "ScarLETH : Design and Control of a Planar Running Robot," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 2–7, 2011.
- [45] Q. Huang, Z. Peng, W. Zhang, L. Zhang, and K. Li, "Design of humanoid complicated dynamic motion based on human motion capture," *2005 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 3536–3541, 2005.
- [46] L. H. Ting, R. Blickhan, and R. J. Full, "Dynamic And Static Stability In Hexapedal Runners," vol. 269, pp. 251–269, 1994.
- [47] G. Lee, K. Seo, S. Lee, J. Park, H. Kim, J. Kim, and T. Seo, "Compliant track-wheeled climbing robot with transitioning ability and high-payload capacity," *2011 IEEE Int. Conf. Robot. Biomimetics*, pp. 2020–2024, Dec. 2011.
- [48] J. W. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," *2013 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 4288–4295, Nov. 2013.
- [49] A. Sproewitz, M. Asadpour, A. Billard, P. Dillenbourg, A. J. Ijspeert, and E. P. F, "Roombots — Modular Robots for Adaptive Furniture," pp. 1–5.

- [50] R. Moeckel, Y. N. Perov, A. T. Nguyen, M. Vespignani, S. Pouya, A. Sproewitz, J. Van Den Kieboom, and A. J. Ijspeert, "Gait Optimization for Roombots Modular Robots - Matching Simulation and Reality," pp. 3265–3272, 2013.
- [51] M. Egerstedt, K. Johansson, J. Lygeros, and S. Sastry, "Behavior Based Robotics Using Regularized Hybrid," *Proc. 3EL Conf. Decis. Control Phoenix, Arizona USA*, no. December, pp. 3400–3405, 1999.
- [52] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," *Proceedings, IEEE Aerosp. Conf.*, vol. 5, pp. 5–2025–5–2036.
- [53] P. Biber, U. Weiss, M. Dorna, and A. Albert, "Navigation System of the Autonomous Agricultural Robot 'BoniRob' *," 2010.
- [54] A. Shirkhodaie and R. Amrani, "Visual Terrain Perception Modeling of Space Planetary Robotic Systems Based on Soft Computing Classifiers *," no. 124858, pp. 42–47, 2005.
- [55] R. Jitpakdee and T. Maneewarn, "Neural networks terrain classification using Inertial Measurement Unit for an autonomous vehicle," *2008 SICE Annu. Conf.*, pp. 554–558, Aug. 2008.
- [56] J.-Y. Park, G.-M. Lee, H. Shim, H. Baek, P.-M. Lee, B.-H. Jun, and J.-Y. Kim, "Head alignment of a single-beam scanning sonar installed on a multi-legged underwater robot," *2012 Ocean.*, pp. 1–5, Oct. 2012.
- [57] X. Li and L. Yang, "Design and Implementation of UAV Intelligent Aerial Photography System," *2012 4th Int. Conf. Intell. Human-Machine Syst. Cybern.*, pp. 200–203, Aug. 2012.
- [58] Q. Chen, Ü. Özgüner, and K. Redmill, "Developing a Completely Autonomous Vehicle," in *Intelligent Systems, IEEE (Volume: 19, Issue: 5)*, Intelligent Transportation Systems, 2004, pp. 8 – 11.
- [59] Z. Bin, A. B. U. Bakar, F. Of, E. Engineering, and S. D. Ehsan, "Classification of Parkinson's Disease Based on Multilayer Perceptron Neural Network," no. June, 2010.
- [60] F. Yaghoubi, A. Ayatollahi, and R. Soleimani, "Classification of Cardiac Abnormalities Using Reduced Features of Heart Rate Variability Signal," vol. 6, no. 11, pp. 1547–1554, 2009.

- [61] S. R. Amendolia, G. Cossu, M. L. Ganadu, B. Golosio, G. L. Masala, and G. M. Mura, "A comparative study of K-Nearest Neighbour, Support Vector Machine and Multi-Layer Perceptron for Thalassemia screening," *Chemom. Intell. Lab. Syst.*, vol. 69, no. 1–2, pp. 13–20, Nov. 2003.
- [62] J. Zhang, H. A. I. Zhao, and B. Lu, "A Comparative Study On Two Large-Scale Hierarchical Text Classification Tasks ' Solutions," no. July, pp. 11–14, 2010.
- [63] S. S. Salankar and B. M. Patre, "Neural Network and Decision Tree Induction: A Comparison in the Domain of Classification of Sonar Signal," *2008 First Int. Conf. Emerg. Trends Eng. Technol.*, pp. 595–600, 2008.
- [64] B. M. Faria, L. P. Reis, N. Lau, and G. Castillo, "Machine Learning algorithms applied to the classification of robotic soccer formations and opponent teams," *2010 IEEE Conf. Cybern. Intell. Syst.*, pp. 344–349, Jun. 2010.
- [65] R. Gopalapillai, J. Vidhya, D. Gupta, and S. Tsb, "Classification of Robotic Data using Artificial Neural Network," pp. 333–337, 2013.
- [66] H. Seraji and a. Howard, "Behavior-based robot navigation on challenging terrain: A fuzzy logic approach," *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 308–321, Jun. 2002.
- [67] a. Howard, H. Seraji, and E. Tunstel, "A rule-based fuzzy traversability index for mobile robot navigation," *Proc. 2001 ICRA. IEEE Int. Conf. Robot. Autom. (Cat. No.01CH37164)*, vol. 3, pp. 3067–3071, 2001.
- [68] H. Seraji, "Rule-based traversability indices for multi-scale terrain assessment," *Proc. 2003 IEEE Conf. Control Appl. 2003. CCA 2003.*, vol. 2, pp. 1469–1476.
- [69] K. Iagnemma and S. Dubowsky, "Terrain estimation for high-speed rough-terrain autonomous vehicle navigation," no. 617.
- [70] K. Iagnemma, H. Shibly, and S. Dubowsky, "On-line terrain parameter estimation for planetary rovers," *Proc. 2002 IEEE Int. Conf. Robot. Autom. (Cat. No.02CH37292)*, vol. 3, no. May, pp. 3142–3147, 2002.
- [71] D. Golda, K. Iagnemma, and S. Dubowsky, "Probabilistic modeling and analysis of high-speed rough-terrain mobile robots," *IEEE Int. Conf. Robot. Autom. 2004. Proceedings. ICRA '04. 2004*, pp. 914–919 Vol.1, 2004.

- [72] J. Huang, J. Lu, and C. X. Ling, "Comparing naive Bayes, decision trees, and SVM with AUC and accuracy," *Third IEEE Int. Conf. Data Min.*, pp. 553–556, 2003.
- [73] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1–2, pp. 169–186, Sep. 2003.
- [74] M. Molavi, J. Bin Yunus, and E. Akbari, "Comparison of Different Methods for Emotion Classification," *2012 Sixth Asia Model. Symp.*, pp. 50–53, May 2012.
- [75] S. Fathima and N. Hundewale, "Comparison of classification techniques-SVM and naives bayes to predict the Arboviral disease-Dengue," *2011 IEEE Int. Conf. Bioinforma. Biomed. Work.*, pp. 538–539, Nov. 2011.
- [76] K. Kim, K. Ko, W. Kim, S. Yu, and C. Han, "Performance comparison between neural network and SVM for terrain classification of legged robot," *SICE Annu. Conf. 2010*, no. 2004, pp. 1343–1348, 2010.
- [77] J. Flórez, F. Calderón, and C. Parra, "Servo load analysis for the classification of surface of locomotion in a modular snake-like robot," pp. 1–6, 2012.
- [78] C. Weiss, N. Fechner, M. Stark, and A. Zell, "Comparison of Different Approaches to Vibration-based Terrain Classification," pp. 1–6.
- [79] R. E. Corresponding, A. Rezaei, and B. Minaei-bidgoli, "Comparison of Classification Methods Based on the Type of Attributes and Sample Size."
- [80] P. Drew, L. Bottaci, G. S. Duthie, and J. R. T. Monson, *Introduction to neural networks, The Lancet Volume*, vol. 350. 1997.
- [81] A. Ben-hur and J. Weston, "Data Mining Techniques for the Life Sciences," vol. 609, pp. 223–239, 2010.
- [82] M. M. Crawford and J. Ghosh, "Integrating support vector machines in a hierarchical output space decomposition framework," *IEEE Int. IEEE Int. IEEE Int. Geosci. Remote Sens. Symp. 2004. IGARSS '04. Proceedings. 2004*, vol. 2, pp. 949–952, 2004.
- [83] K. Li, Y. Chen, and H. Liu, "A new method of evolving hardware design based on IIC bus and AT24C02," *Proc. 10th World Congr. Intell. Control Autom.*, pp. 104–107, Jul. 2012.

- [84] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 4, pp. 1026–7, Jan. 2002.
- [85] J. Gausemeierl, C. Y. Low, D. Steffen, S. Deyterl, and E. D. Stefifen, "Specifying the Principle Solution in Mechatronic Development Enterprises," *IEEE Int. Syst. Conf. SysCon 2008*, 2008.

APPENDICES

APPENDIX A

1. C. Low, K. Azmi, N. Ayuni, F. Rosdayanti, A. Syahmi. **Steering Behavior of a Track-Driven Paintball Robot.** *International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).*
2. K. Azmi, SC. Tan, CY. Low, A. Tan, J. Ahmed. **Terrain Classification for Track-Driven Agricultural Robots.** *2nd International Conference on System-integrated Intelligence sysInt2014. New Challenges for Product and Production Engineering.*
3. K. Azmi, H. Hamli, CY. Low, A. Jaffar, E. Salleh. **Stair Climbing of a Track-Driven Mobile Robot with Flipper Arm.** *Applied Mechanics and Materials Vol. 393 (2013) pp 586-591 Trans Tech Publications.*
4. CY. Low, M. Stecklina; K. Azmi, SC. Tan, R. Dumitrescu, A. Jaffar. **Tracked-driven Agricultural Robot with Terrain Classification Functionality.** *Special Issue on System-Integrated Intelligence of the Mechatronics Journal.*

APPENDIX B

Coding for Data Acquisition

```
int x = analogRead(xpin);
int y = analogRead(ypin);
int z = analogRead(zpin);
float zero_G = 512.0;
float scale = 102.3;
Serial.print(((float)z - zero_G)/scale);
Serial.print("\n");
if (rpmcount1 >= 20 || rpmcount2 >=20)
    rpm1 = 30*10/(millis() - timeold1)*rpmcount1;
    timeold1 = millis();
    rpmcount1 = 0;
    rpm2= 30*10/(millis() - timeold2)*rpmcount2;
    timeold2 = millis();
    rpmcount2 = 0
void rpm_fun1()
    rpmcount1++;
void rpm_fun2()
    rpmcount2++;
```

The Arduino source code shown is for the motion of the track robot

```
int button1 = 2;
int button2 = 3;
int button3 = 4;
```

```
int dir1 = 8;
int pwm1 = 10;
int dir2 = 13;
int pwm2 = 9;
```

```
int val1 = 0;
int val2 = 0;
int val3 = 0;
```

```
int rpm1;
int rpm2;
int rpm3;
int rpm4;
```

```
////////state2
int a = 123; // rpm motor actual
int b = 123;
```

```
////////state3
int c = 67; //right
int d = 123; //left
```

```
void setup()
{
  Serial.begin(9600);
  Serial.print("READY");
  pinMode(button1, INPUT);
  pinMode(button2, INPUT);
  pinMode(button3, INPUT);
  pinMode(dir1, OUTPUT);
  pinMode(pwm1, OUTPUT);
  pinMode(dir2, OUTPUT);
  pinMode(pwm2, OUTPUT);
}
```

```
a = map(a, 0, 170, 0, 255); //right
b = map(b, 0, 170, 0, 255); //left
c = map(c, 0, 170, 0, 255); //right
d = map(d, 0, 170, 0, 255); //left }
```

```
void loop()
{
  val1 = digitalRead(button1);
  val2 = digitalRead(button2);
  val3 = digitalRead(button3);
  if (val2 == HIGH)
```

```
{
  Serial.println(" HIGH ");
  initial();
  delay(3000);
  beforeReduce();
  delay(3000);
  afterReduce();
  delay(27000); }
else
{
  initial(); }
void initial()
```

```
{
  digitalWrite(dir1, HIGH);
  analogWrite(pwm1, 0);
  digitalWrite(dir2, HIGH);
  analogWrite(pwm2, 0); //PWM Speed Control
}
```

```
void beforeReduce()
{
  digitalWrite(dir1, HIGH);
  analogWrite(pwm1, 170); // Speed Control RIGHT
  digitalWrite(dir2, HIGH);
  analogWrite(pwm2, 170); //Speed Control LEFT }
```

```
void afterReduce() {
  digitalWrite(dir1, HIGH);
  analogWrite(pwm1, 85); //Speed Control RIGHT
  digitalWrite(dir2, HIGH); analogWrite(pwm2, 170); // Speed Control LEFT }
```

Data Acquired from the Robot

Sands (ms ⁻²)	Gravels (ms ⁻²)	Soil (ms ⁻²)	Time (milliseconds)
-0.98	-0.98	-0.97	10
-0.94	-0.95	-1	20
-0.85	-0.96	-0.77	30
-0.99	-0.96	-0.96	40
-1	-0.83	-1	50
-0.98	-1.06	-1.01	60
-0.93	-0.72	-0.97	70
-1.05	-0.67	-0.84	80
-1.02	-0.9	-0.84	90
-1.02	-0.97	-1.08	100
-1	-0.95	-1.38	110
-0.92	-1.06	-0.98	120
-0.98	-1.11	-0.93	130
-1.05	-0.97	-1.23	140
-1	-1.15	-0.84	150
-1.05	-1.02	-0.78	160
-0.9	-0.94	-1.08	170
-1.01	-0.98	-0.92	180
-0.95	-0.97	-0.94	190
-1.06	-0.92	-0.97	200
-1	-1.13	-0.89	210
-0.95	-0.73	-0.94	220
-1.04	-1.07	-1.06	230
-0.93	-0.65	-0.9	240
-0.94	-1.05	-0.79	250
-1.03	-1	-1.22	260
-0.94	-0.95	-1.12	270
-0.94	-0.92	-0.93	280
-0.94	-0.97	-1.34	290
-0.9	-1	-0.81	300
-1.06	-0.8	-0.85	310
-1.03	-0.94	-1.11	320
-0.96	-0.97	-1.02	330
-1.03	-0.93	-0.89	340
-1	-1	-0.86	350
-1.01	-0.82	-1.06	360
-1.01	-0.7	-1.18	370
-0.92	-1.1	-0.93	380
-0.96	-1.16	-0.98	390
-1.02	-0.93	-0.79	400
-0.9	-0.91	-0.73	410
-0.96	-1.23	-0.72	420
-1.03	-1.19	-1.29	430

-0.89	-0.77	-1.14	440
-0.89	-1.01	-1.04	450
-0.96	-0.87	-1.09	460
-1.05	-0.92	-0.83	470
-1.03	-0.96	-1.13	480
-0.87	-1.09	-1.1	490
-0.94	-1.07	-0.85	500
-0.95	-1.03	-1.13	510
-0.97	-0.84	-0.77	520
-1.06	-0.76	-0.71	530
-1	-1.04	-1.13	540
-0.92	-1.16	-0.96	550
-1.03	-0.68	-0.97	560
-0.94	-0.91	-0.84	570
-1.04	-0.96	-0.86	580
-1.01	-1.34	-0.84	590
-0.93	-0.57	-0.86	600
-1.09	-0.67	-0.96	610
-0.97	-1.21	-0.78	620
-1.01	-1.08	-0.83	630
-1.08	-0.84	-1.09	640
-0.95	-1.04	-1.02	650
-1.04	-1.1	-0.97	660
-1.04	-0.72	-0.85	670
-1.04	-0.84	-1.35	680
-0.96	-0.89	-0.99	690
-0.97	-0.58	-0.84	700
-1.06	-1.18	-0.93	710
-1	-0.65	-0.84	720
-1.01	-0.84	-0.97	730
-0.95	-1.09	-0.88	740
-0.94	-0.74	-1.46	750
-1.05	-1.01	-0.8	760
-1.01	-1.08	-0.86	770
-0.93	-0.59	-0.77	780
-1.03	-1.07	-0.94	790
-1.01	-1.07	-0.81	800
-0.95	-0.78	-1.25	810
-1.07	-0.87	-1.04	820
-0.98	-1.15	-1.14	830
-1	-0.88	-0.89	840
-1	-0.74	-1.02	850
-0.92	-1.25	-1.17	860
-0.87	-0.84	-0.94	870
-1.01	-1.38	-0.84	880

-1.12	-0.87	-0.99	890
-0.98	-0.94	-0.99	900
-0.97	-0.93	-0.91	910
-0.97	-1.2	-0.65	920
-0.99	-0.93	-1.22	930
-0.99	-1	-1.35	940
-0.99	-0.77	-0.88	950
-0.99	-1.41	-0.83	960
-0.98	-0.99	-1.05	970
-0.98	-0.76	-0.9	980
-0.99	-1.38	-1.02	990
-0.99	-0.72	-1.02	1000

Sands (ms ⁻²)	Gravels (ms ⁻²)	Soil (ms ⁻²)	Time (milliseconds)
0	0	0	0
37	20	23	10
37	20	23	20
37	25	41	30
37	25	41	40
43	31	47	50
43	31	56	60
40	36	62	70
40	36	59	80
50	31	59	90
62	31	46	100
65	27	59	110
65	27	62	120
65	32	73	130
62	56	76	140
59	70	62	150
62	70	62	160
56	46	67	170
62	46	65	180
59	41	65	190
56	41	53	200
56	26	73	210
62	26	73	220
67	26	73	230
84	22	73	240
65	22	67	250
65	46	67	260
65	46	70	270
67	47	73	280
65	59	73	290
65	62	62	300
50	56	67	310
70	56	70	320
67	46	90	330
67	62	79	340
70	65	70	350
62	65	67	360
59	65	65	370
62	43	56	380
62	43	59	390
65	41	76	400
65	41	70	410
56	50	67	420

65	67	73	430
67	62	70	440
79	62	62	450
79	42	65	460
67	42	67	470
67	40	67	480
67	40	70	490
65	36	62	500
65	65	65	510
50	62	65	520
73	59	73	530
70	59	96	540
67	50	70	550
67	67	73	560
67	59	73	570
62	62	67	580
67	59	62	590
65	65	76	600
73	65	76	610
70	56	76	620
62	56	73	630
67	37	76	640
76	37	62	650
84	44	65	660
76	70	67	670
67	70	62	680
67	70	67	690
70	42	56	700
65	42	59	710
65	30	62	720
49	30	65	730
65	26	93	740
62	26	67	750
65	47	67	760
65	47	67	770
65	43	70	780
65	59	62	790
0	67	59	800
	67	79	810
	46	73	820
	46	76	830
	42	76	840
	42	70	850
	46	65	860
	62	65	870

	67	70	880
	56	67	890
	62	65	900
	62	65	910
	48	65	920
	84	70	930
	84	93	940
	47	73	950
	56	70	960
	62	70	970
	62	70	980
	62	70	990
	47	52	1000
	47	82	1010
	32	67	1020
	0	0	1030

APPENDIX C

The plane motion of rigid body is the result of continuous rotation about the instantaneous center which fluctuates with time. In this case, the coordinates of the instantaneous center of rotation of the hull are defined by

$$\left[\frac{V_y}{\dot{\theta}}, \frac{V_x}{\dot{\theta}} \right]$$

The instantaneous velocities of the track robot relative to the hull defined as V_{t1} and V_{t2} which correspond to the circumferential velocity of the drive sprockets. Both tracks have no degree of freedom in the y-direction, so it is said that the track move in the speed of the hull in that respective direction. Therefore, the components of slip velocity of the track shoe under the i^{th} road wheel are expressed respectively by these equations;

$$V_{sxj} = V_x \pm \frac{B}{2} \dot{\theta} - V_{tj}$$

$$V_{sxi} = V_y + \left(\frac{i-1}{n-1} - \frac{1}{2} \right) L \dot{\theta}$$

Where subscript I present the order of road wheels from the front of vehicle, and subscript inner track when $j = 1$ and outer track when $j = 2$. The direction of slipping of the track shoe under the (i, j) road wheel on the other hand is expressed as follows:

$$\Psi_{ij} = \cos^{-1} \left(\frac{V_{sxj}}{\sqrt{V_{sxj}^2 + V_{syi}^2}} \right)$$

$$\Psi_{ij} = \sin^{-1} \left(\frac{V_{syj}}{\sqrt{V_{sxj}^2 + V_{syi}^2}} \right)$$

S_j is the ratio that used in the friction coefficient. It is the difference between the track velocity and the forward velocity of the part of the hull on the center line of the track divided by the largest in magnitude of these two velocities.

$$S_j = \frac{V_{sxj}}{\max(V_{tj} + V_{sxj}, V_{tj})}$$

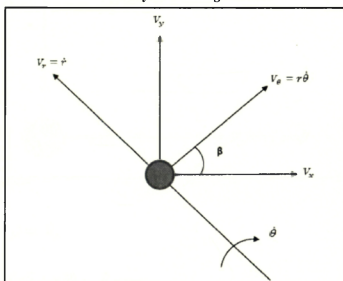
From this equation of slip ratio, the slip ratios are able to obtain with magnitude greater than 1. In order to avoid this problem, S_j is restricted to maximum value of 1.

The coordinates of the instantaneous center of rotation on both tracks is defined by $\left[\frac{V_y}{\dot{\theta}}, \frac{V_z}{\dot{\theta}}\right]$ where the coordinate $\frac{V_y}{\dot{\theta}}$ corresponds to D, the displacement of pivoting point of vehicle, and the coordinate $\frac{V_{sy1}}{\dot{\theta}}$ and $\frac{V_{sy2}}{\dot{\theta}}$ correspond to A_1, A_2 , the clip radii of tracks. The kinematics and dynamics of the individual tracks, hence of the vehicle, are determined by these coordinates of two instantaneous centers of rotations.

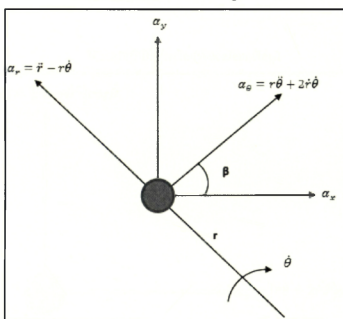
Derivations of Equation for Acceleration

In deriving the equations for acceleration, kinematics of turning particles is taken into consideration.

Velocity of Turning Particle

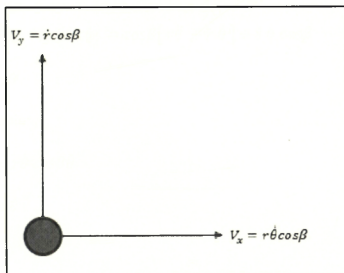


Acceleration of Turning Particle



The particle is rotated to make it coincide with moving vehicle frame in horizontal x -axis and vertical y -axis.

Rotated Particle (Velocity)

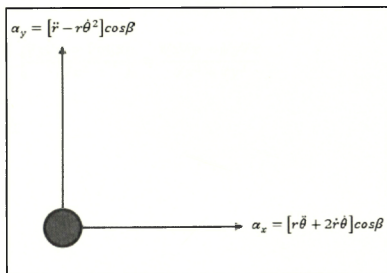


$$\frac{d}{dt}(V_y) = \dot{r} \cos \beta = \dot{V}_y$$

$$\frac{d}{dt}(V_x) = \cos \beta [r \ddot{\theta} + \dot{r} \dot{\theta}] = \dot{V}_x$$

$$a_y = [\ddot{r} - r \dot{\theta}^2] \cos \beta$$

Rotated Particle (Acceleration)



Forward acceleration,

$$\alpha x = \cos\beta [r\ddot{\theta} + \dot{r}\dot{\theta} + \dot{r}\dot{\theta}] = \cos\beta [r\ddot{\theta} + \dot{r}\dot{\theta}] + \dot{r}\dot{\theta}\cos\beta$$

$$\alpha x = \dot{V}_x + V_y\dot{\theta}$$

Lateral acceleration,

$$\alpha y = \dot{r}\cos\beta - r\dot{\theta}\cos\beta\dot{\theta}$$

$$\alpha y = \dot{V}_y - V_x\dot{\theta}$$

Derivation of Equation for Side Slip Ratio

$$\frac{d}{dt}\{\tan^{-1}(x)\} = \frac{1}{1+x^2}$$

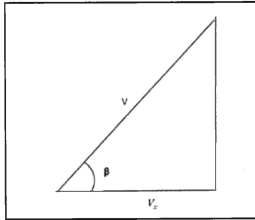
$$\beta = \frac{d}{dt}\beta = \frac{d}{dt}\left(\tan^{-1}\left(\frac{V_y}{V_x}\right)\right)$$

$$= \left[\frac{1}{1 + \frac{V_y}{V_x}} \right] \cdot \frac{d}{dt} \left(\frac{V_y}{V_x} \right)$$

$$= \left[\frac{1}{\frac{V_x^2 + V_y^2}{V_x^2}} \right] \cdot \left[\frac{V_x V_y - V_y V_x}{V_x^2} \right] = \frac{V_x V_y - V_y V_x}{V_x^2 + V_y^2}$$

$$\dot{\beta} = \frac{V_x \dot{V}_y - V_y \dot{V}_x}{V^2}$$

Derivation of Equation for Side Slip Ratio, $\dot{\beta}$



V_x and V_y are function of time;

$$\tan\beta = \left(\frac{V_y}{V_x}\right)$$

$$\beta = \tan^{-1}\left(\frac{V_y}{V_x}\right)$$

$$\dot{\beta} = \frac{d}{dt}\beta = \frac{d}{dt}\left(\tan^{-1}\left(\frac{V_y}{V_x}\right)\right)$$

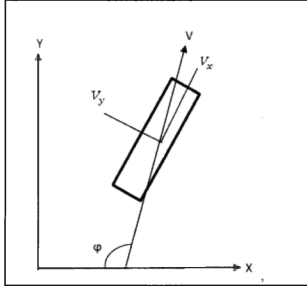
$$= \left[\frac{1}{1 + \frac{V_y^2}{V_x^2}} \right] \cdot \frac{d}{dt}\left(\frac{V_y}{V_x}\right)$$

$$= \left[\frac{1}{\frac{V_x^2 + V_y^2}{V_x^2}} \right] \cdot \left[\frac{V_x V_y - V_y V_x}{V_x^2} \right] = \frac{V_x V_y - V_y V_x}{V_x^2 + V_y^2}$$

$$\dot{\beta} = \frac{V_x \dot{V}_y - V_y \dot{V}_x}{V^2}$$

Derivation of the Coordinates on the Vehicle

(X_t, Y_t) = coordinates fixed on the ground as a function of time



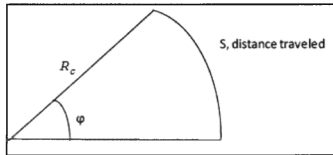
$$\frac{d}{dt} X_t = V \cos (180^\circ - \varphi) = -V \cos \varphi$$

$$X_t = - \int_0^t V \cos \varphi dt$$

$$\frac{d}{dt} Y_t = V \sin (180^\circ - \varphi) = V \sin \varphi$$

$$Y_t = \int_0^t V \sin \varphi dt$$

Derivation of Equation for Radius of Curvature



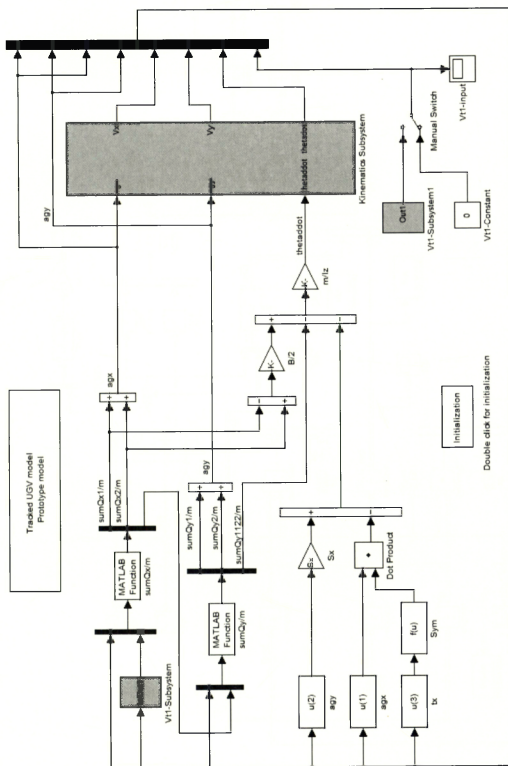
$$S = R_c \varphi$$

$$V = \frac{ds}{dt} = R_c \dot{\varphi}$$

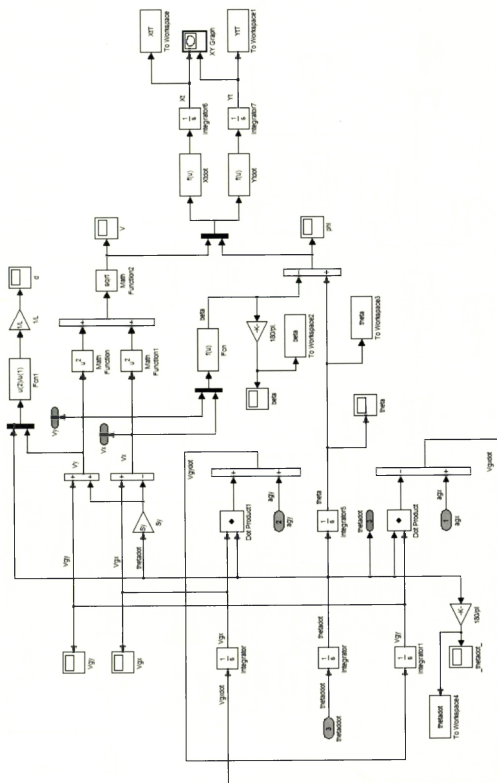
$$R_c = \frac{v}{\dot{\varphi}}$$

APPENDIX D

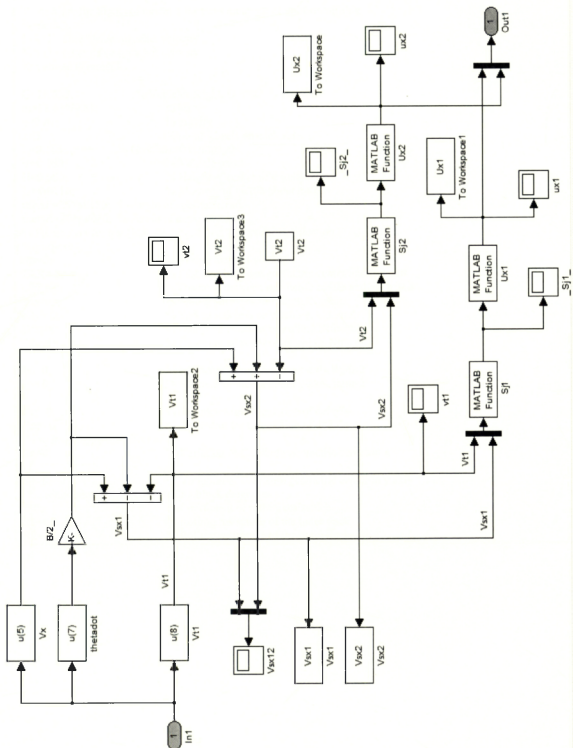
The main system of the prototype tracked vehicle



The kinematic subsystem



The Vt subsystem



APPENDIX E

This is the m-file programming to verify the parameters of the prototype tracked vehicle model. The m-file is used as an input and able to manipulate parameter changed for different results.

```
global Vt2 Sx Sy L B G m g n H DhR DhP Iz Iy Ix Ktx Kty Kz Bw f R1 R2 ms k E1 E2 thetaF thetaR

% these are the inputs of model 1 for the simulation of uniform turning
% motion.
% Vt1= inner drive sprocket velocity
% Vt2= outer drive sprocket velocity
% Vehicle Parameters

G=18.639;           % Vehicle weight, G=mg
m=18.639/9.81;      % kg
g=9.81;             % gravity
n=2;                % no of wheels on each side
L=0.185;            % Ground contact length
B=0.189;            % Vehicle tread
H=0.060;            % Height of center of gravity
Iz=0.0111;          % Moment of inertia about z-axis
thetaF=25*pi/180;   % Angle of approach, rad
thetaR=27*pi/180;   % Angle of departure, rad
% Parameters for transient steering, all set to zero for verification
% purpose
ms=0;               % Spring mass of vehicle
DhR= 0;             % Distance between geometric center and roll axis
DhP= 0;             % Distance between geometric center and pitch axis
Ktx= 0;             % Spring stiffness on rolling, Nm/rad
Kty= 0;             % Spring stiffness on pitching, kgm/rad
Dtx= 0;             % Damping constant on rolling, Nms/rad
Dty= 0;             % Damping constant on pitching, Nms/rad

Ix= 0;              % Moment of inertia about x-axis, kgms2
Iy= 0;              % Moment of inertia about y-axis, kgms2

Kz= 0;              % Parameter for calculation of deltaSy
Bw= 0;              % Parameter for calculation of deltaSy
% to investigate the effect of cg
Sx=0;               % Variation of C.G from geometric center in x
                    axis
Sy=0;               % Variation of C.G from geometric center in y axis
% Assumptions while constructing block diagrams
f=0;                % Coefficient of rolling resistance
R1=0;               % Rolling resistance of the inner track
R2=0;               % Rolling resistance of the outer track
```

APPENDIX F

The m-file listed is the function block to calculate the sum of longitudinal frictional forces on each wheel.

```
function sum_Qx=QxnitiTrackT(u)
global Vt2 Sx Sy Uo L B G m g n H DhR DhP Iz Iy Ix Ktx Kty Kz Bw f R I R2 thetaF thetaR

% This is to calculate the sum of Qxi1 and Qxi2 from i=1 to n=5
% The effect of track tension is neglected

%Vx =u(5);
Vy =u(6);
w =u(7);

agx =u(1);
agy =u(2);

Vt1 =u(8);

tx =u(3);
ty =u(4);

Ux1 =u(9);
Ux2 =u(10);

deltaSy = Bw*(1-exp(Kz*tx));

Vsx1 = Vx - B*w/2 - Vt1 + deltaSy*w;
Vsx2 = Vx + B*w/2 - Vt2 + deltaSy*w;

HR = H-DhR;
HP = H-DhP;

Py1 = +agy/(n*B)*(HR+DhR*cos(tx)-Sy*sin(tx)); % different sign for different paper!!!!!!
Py2 = -agy/(n*B)*(HR+DhR*cos(tx)-Sy*sin(tx));

for i=1:n

    Vsy(i) = Vy + ((i-1)/(n-1)-1/2)*L*w;

    if sqrt(Vsx1^2+Vsy(i).^2)==0
        cosi(i)=-1;
    else
        cosi1(i) = -Vsx1/sqrt(Vsx1^2+Vsy(i).^2);
    end
    if sqrt(Vsx2^2+Vsy(i).^2)==0

        cosi2(i)=-1;
    else
        cosi2(i) = -Vsx2/sqrt(Vsx2^2+Vsy(i).^2);
    end

Term1(i)= g*( 1/(2*n) - 3*(2*i-n-1)*(Sx*cos(ty)-DhP*sin(ty)) / (n*(n+1)*L) );

Pw1(i) = Term1(i)*( 1 - 2/B*(Sy*cos(tx)+DhR*sin(tx)) );
Pw2(i) = Term1(i)*( 1 + 2/B*(Sy*cos(tx)+DhR*sin(tx)) );

Px(i) = 3*(2*i-n-1)*(HP+DhP*cos(ty)-Sx*sin(ty)) / (n*(n+1)*L) * agx;

Pi1(i) = Pw1(i) + Py1 + Px(i);
Pi2(i) = Pw2(i) + Py2 + Px(i);
```

```

    Qx1(i) = Pi1(i)*Ux1*cosi1(i);
    Qx2(i) = Pi2(i)*Ux2*cosi2(i);
end

if Vsx1 <= 0
    t11 = 0;
    t21 = 0;
    t31 = 0;
    t41 = 0;
    % if Vt2 < 6
    % tn1 = - sin(thetaR)*sum(Qx1); %%% low speed -ve
    % else
    tn1 = sin(thetaR)*sum(Qx1); %%% high speed +ve (10m/s)
    % end
elseif Vsx1 > 0
    t11 = sin(thetaF)*sum(Qx1);
    t21 = 0;
    t31 = 0;
    t41 = 0;
    tn1 = 0;
end
t1(1)=t11;
t1(2)=t21;
t1(3)=t31;
t1(4)=t41;
t1(5)=tn1;
if Vsx2 <= 0
    t12 = 0;
    t22 = 0;
    t32 = 0;
    t42 = 0;
    % if Vt2 < 6
    % tn2 = - sin(thetaR)*sum(Qx2); %%% low speed -ve
    % else
    tn2 = sin(thetaR)*sum(Qx2); %%% high speed +ve (10m/s)
    % end
elseif Vsx2 > 0
    t12 = sin(thetaF)*sum(Qx2);
    t22 = 0;
    t32 = 0;
    t42 = 0;
    tn2 = 0;
end

```

```

t2(1)=t12;
t2(2)=t22;
t2(3)=t32;
t2(4)=t42;
t2(5)=tn2;
for q=1:5;
    PM1(q) = (5*t11-tn1+3*t12-3*tn2)/(2*n) + 3*(tn1+tn2-t11-t12)*q/(n*(n+1)) - t1(q);
    PM2(q) = (3*t11-3*tn1+5*t12-tn2)/(2*n) + 3*(tn1+tn2-t11-t12)*q/(n*(n+1)) - t2(q);

    PPi1(q) = Pi1(q) + PM1(q);
    PPi2(q) = Pi2(q) + PM2(q);

    PQx1(q) = PPi1(q)*Ux1*cosi1(q);
    PQx2(q) = PPi2(q)*Ux2*cosi2(q);
    end
    sumQx1=sum(PQx1);
    sumQx2=sum(PQx2);

    sum_Qx=[sumQx1,sumQx2,PPi1,PPi2];

```

APPENDIX G

The m-file listed is the function block to calculate the sum of lateral frictional forces on each wheel.

```
function sum_Qy=QynitiTrackT(u)
global Vt2 Sx Sy L B G m g n H DhR DhP Iz Iy Ix Ktx Kty Kz Bw f R1 R2 k E1 E2 thetaF thetaR

% This is to calculate the sum of Qy11 and Qy12 and Qy1122 from i=1 to n=5
% The effect of track tension is neglected

agx =u(1);
agy =u(2);
tx =u(3);
ty =u(4);
Vx =u(5);
Vy =u(6);
w =u(7);
Vt1 =u(8);

PPi1(1)=u(9);
PPi1(2)=u(10);
PPi1(3)=u(11);
PPi1(4)=u(12);
PPi1(5)=u(13);

PPi2(1)=u(14);
PPi2(2)=u(15);
PPi2(3)=u(16);
PPi2(4)=u(17);
PPi2(5)=u(18);

Uy = k*E1*(1-exp(-E2));

deltaSy = Bw*(1-exp(Kz*tx));

Vsx1 = Vx - B*w/2 - Vt1 + deltaSy*w;
Vsx2 = Vx + B*w/2 - Vt2 + deltaSy*w;

for i=1:n

    Vsy(i) = Vy + ((i-1)/(n-1)-1/2)*L*w;

    if sqrt(Vsx1^2+Vsy(i).^2)==0
        sini1(i)=0;
    else
        sini1(i) = -Vsy(i)/sqrt(Vsx1^2+Vsy(i).^2);
    end

    if sqrt(Vsx2^2+Vsy(i).^2)==0
        sini2(i)=0;
    else
        sini2(i) = -Vsy(i)/sqrt(Vsx2^2+Vsy(i).^2);
    end

    Qy1(i) = PPi1(i)*Uy*sini1(i);
    Qy2(i) = PPi2(i)*Uy*sini2(i);

    Qy11(i) = ( 1/2 - (i-1)/(n-1) ) *L* Qy1(i);
    Qy22(i) = ( 1/2 - (i-1)/(n-1) ) *L* Qy2(i);
end
sumQy1=sum(Qy1);
sumQy2=sum(Qy2);
sumQy1122=sum(Qy11)+sum(Qy22);

sum_Qy=[sumQy1,sumQy2,sumQy1122];
```

APPENDIX H

The m-file shown is the function block to calculate the slip track ratio.

```
function Sj=SlipRatio(u)

Vtj=u(1);
Vsxj=u(2);
data=[(Vtj+Vsxj),Vtj];
[extr,l]=max([abs(Vtj+Vsxj),abs(Vtj)]);

if data(l)<0
    extr=-extr;
end

Sj=Vsxj/extr;

if Sj<=1
    Sj=-1;
elseif Sj>1
    Sj=1;
end
```

APPENDIX I

The Arduino source code shown is for the compass and DC motor of the prototype tracked vehicle

```
#include <Wire.h>
```

```
int E1 = 6;
```

```
int M1 = 7;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  Wire.begin();
```

```
  pinMode(M1, OUTPUT);
```

```
  pinMode(E1, OUTPUT); }
```

```
void loop()
```

```
{
```

```
  Wire.beginTransmission(0x21);
```

```
  Wire.write("A"); // Send "Get Data" command,
```

```
  info from Datasheet
```

```
  Wire.requestFrom(0x21, 2); //get the two data bytes,
```

```
  MSB and LSB
```

```
  byte MSB = Wire.read(); // Result will be in tenths  
  of degrees (0 to 3599)
```

```
  byte LSB = Wire.read(); // Provided in binary  
  format over two bytes."
```

```
  Wire.endTransmission();
```

```
  float myres = ((MSB << 8) + LSB) / 10;
```

```
  if (myres >= 0 && myres <= 5 || myres >= 355 &&  
  myres <= 360 )
```

```
  {
```

```
    digitalWrite(M1,LOW);
```

```
    analogWrite(E1,0); }
```

```
  else if (myres >= 3 && myres <= 10 )
```

```
  {
```

```
    digitalWrite(M1,HIGH);
```

```
    analogWrite(E1,20); }
```

```
  else if (myres >= 3 && myres <= 20 )
```

```
  {
```

```
    digitalWrite(M1,HIGH);
```

```
    analogWrite(E1,150); }
```

```
  else if (myres >= 3 && myres <= 60 )
```

```
  {
```

```
    digitalWrite(M1,HIGH);
```

```
    analogWrite(E1,150); }
```

```
  else if (myres >= 3 && myres <= 100 )
```

```
  {
```

```
    digitalWrite(M1,HIGH);
```

```
    analogWrite(E1,255); }
```

```
  else if (myres >= 3 && myres <= 180 )
```

```
  {
```

```
    digitalWrite(M1,HIGH);
```

```
    analogWrite(E1,255); }
```

```
  else if (myres > 347 && myres <= 357)
```

```
  {
```

```
    digitalWrite(M1,LOW);
```

```
    analogWrite(E1,20); }
```

```
  else if (myres > 320 && myres <= 357)
```

```
  {
```

```
    digitalWrite(M1,LOW);
```

```
    analogWrite(E1,150); }
```

```
  else if (myres > 180 && myres <= 357)
```

```
  {
```

```
    digitalWrite(M1,LOW );
```

```
    analogWrite(E1,150);
```

```
  }
```

APPENDIX J

The Arduino source code shown is for the encoder and ultrasonic-range finder of the prototype tracked vehicle

```
volatile byte rpmcount1,rpmcount2;
unsigned int rpm1,rpm2;
unsigned long timeold1,timeold2;
#include <Servo.h>
#include <Wire.h>
int speed1;
int speed2;
float val , val1;
int dist, dist1;
int E1 = 12;
int M1 = 11;
const int pwPin = 9;
const int pwPin1 = 10;
long pulse, inches, pulse1, inches1, cm1, cm;
void setup()
{Serial.begin(9600);
Wire.begin();
Serial.print("READY");
pinMode(M1, OUTPUT);
pinMode(E1, OUTPUT);
pinMode(pwPin, INPUT);
pinMode(pwPin1, INPUT);
digitalWrite(19,HIGH);
digitalWrite(18,HIGH);
attachInterrupt(4, rpm_fun1, RISING);
attachInterrupt(5, rpm_fun2, RISING);
rpmcount1 = 0;
rpm1 = 0;
timeold1 = 0;
rpmcount2 = 0;
rpm2 = 0;
timeold2 = 0; }
void loop(){
Wire.beginTransmission(0x21);
Wire.write("A"); // Send "Get Data" command, info
from Datasheet
//delay(100); // interface command delay
Wire.requestFrom(0x21, 2); //get the two data bytes,
MSB and LSB
byte MSB = Wire.read(); // Result will be in tenths of
```

```
degrees (0 to 3599)
byte LSB = Wire.read(); // Provided in binary format
over two bytes."
Wire.endTransmission();
float myres = ((MSB << 8) + LSB) / 10;
//Used to read in the pulse that is being sent by the
MaxSonar device.
//Pulse Width representation with a scale factor of 147
uS per Inch.
pulse = pulseIn(pwPin, HIGH);
pulse1 = pulseIn(pwPin1, HIGH); //147uS per inch
inches = pulse/147;
inches1 = pulse1/147; //change inches to centimetres
cm = inches * 2.54;
cm1 = inches1 * 2.54;
Serial.print(cm);
Serial.print("\t");
Serial.print(cm1);
Serial.print("\t");
if (rpmcount1 >= 20 || rpmcount2 >=20) {
//Update RPM every 20 counts, increase this for better
RPM resolution,
//decrease for faster update
rpm1 = 30*10/(millis() - timeold1)*rpmcount1;
timeold1 = millis();
rpmcount1 = 0;
rpm2= 30*10/(millis() - timeold2)*rpmcount2;
timeold2 = millis();
rpmcount2 = 0;
speed1 = map(rpm1, 0, 470, 0, 190);
speed2 = map(rpm2, 0, 470, 0, 190);
Serial.print(speed2);///left
Serial.print("\t");
Serial.println(speed1);///right
else{Serial.print(" ");
Serial.println(0); } }
void rpm_fun1()
{rpmcount1++;//Each rotation, this interrupt }
void rpm_fun2()
{ rpmcount2++; }
```

APPENDIX K

```

function [nsv, alpha, b0] = svc(X,Y,ker,C)
%SVC Support Vector Classification
%%
%% Usage: [nsv alpha bias] = svc(X,Y,ker,C)
%%
%% Parameters: X    - Training inputs
%%              Y    - Training targets
%%              ker   - kernel function
%%              C     - upper bound (non-separable case)
%%              nsv   - number of support vectors
%%              alpha - Lagrange Multipliers
%%              b0    - bias term
%%

if (nargin < 2 | nargin > 4) % check correct number of arguments
    help svc
else

    fprintf('Support Vector Classification\n')
    fprintf('_____ \n')
    n = size(X,1);
    if (nargin < 4) C=Inf, end
    if (nargin < 3) ker='linear', end

    % tolerance for Support Vector Detection
    epsilon = svtol(C);

    % Construct the Kernel matrix
    fprintf('Constructing ... \n');
    H = zeros(n,n);
    for i=1:n
        for j=1:n
            H(i,j) = Y(i)*Y(j)*svkernel(ker,X(i,:),X(j,:));
        end
    end
    c = -ones(n,1);

    % Add small amount of zero order regularisation to
    % avoid problems when Hessian is badly conditioned.
    H = H+1e-10*eye(size(H));

    % Set up the parameters for the Optimisation problem

```

```

vlb = zeros(n,1); % Set the bounds: alphas >= 0
vub = C*ones(n,1); % alphas <= C
x0 = zeros(n,1); % The starting point is [0 0 0 0]
neqcstr = nobias(ker); % Set the number of equality constraints (1 or 0)
if neqcstr
    A = Y'; b = 0; % Set the constraint Ax = b
else
    A = []; b = [];
end

% Solve the Optimisation Problem

fprintf('Optimising ...\n');
st = cputime;

[alpha lambda how] = qp(H, c, A, b, vlb, vub, x0, neqcstr);

fprintf('Execution time: %4.1f seconds\n', cputime - st);
fprintf('Status : %s\n', how);
w2 = alpha'*H*alpha;
fprintf('|w0|^2 : %f\n', w2);
fprintf('Margin : %f\n', 2/sqrt(w2));
fprintf('Sum alpha : %f\n', sum(alpha));

% Compute the number of Support Vectors
svi = find( alpha > epsilon);
nsv = length(svi);
fprintf('Support Vectors : %d (%3.1f%%)\n', nsv, 100*nsv/n);

% Implicit bias, b0
b0 = 0;
% Explicit bias, b0
if nobias(ker) ~= 0
    % find b0 from average of support vectors on margin
    % SVs on margin have alphas: 0 < alpha < C
    svii = find( alpha > epsilon & alpha < (C - epsilon));
    if length(svii) > 0
        b0 = (1/length(svii))*sum(Y(svii) - H(svii,svi)*alpha(svi).*Y(svii));
    else
        fprintf('No support vectors on margin - cannot compute bias.\n');
    end
end
end

```

Coding to Plot Classification

```
clear all;
% Read a xls or.xlsx file
[TrainData,txt_train] = xlsread('Balloon.xlsx');

% Read a xls or.xlsx file
[TestData,txt_test] = xlsread('BalloonTest.xlsx');

kernelType = 2; % Set kernel type: 0 = Linear; 1 = Quadratic function; 2 = RBF; 3 = Polynomial
svmStruct = [];
% train a SVM classifier with a chosen kernel function
if (kernelType == 0)
    % Linear Kernel
    svmStruct = svmtrain(TrainData(:,2:end),TrainData(:,1));
elseif (kernelType == 1)
    svmStruct = svmtrain(TrainData(:,2:end),TrainData(:,1),'Kernel_Function','quadratic');
elseif (kernelType == 2)
    svmStruct = svmtrain(TrainData(:,2:end),TrainData(:,1),'Kernel_Function','rbf','RBF_Sigma',1);
%default setting is 1, range between 0 and 1.
else
    svmStruct = svmtrain(TrainData(:,2:end),TrainData(:,1),'Kernel_Function','polynomial','Polyorder',3);
%default setting is 3, can be >3
end

% evaluate accuracy of the SVM classifier based on training set
train_predictclasses = svmclassify(svmStruct,TrainData(:,2:end));
train_accuracy = sum(TrainData(:,1)==train_predictclasses)/length(TrainData(:,1))*100

% evaluate accuracy of the SVM classifier based on test set
test_predictclasses = svmclassify(svmStruct,TestData(:,2:end));
test_accuracy = sum(TestData(:,1)==test_predictclasses)/length(TestData(:,1))*100
```